



TESIS - SS14 2501

***MULTI OUTPUT NEURAL NETWORK* UNTUK
PERAMALAN TEMPERATUR DI SEMARANG
DENGAN PENDEKATAN BAYESIAN**

ZUBDATU ZAHRATI
NRP. 1314 201 037

Dosen Pembimbing
Dr. Kartika Fithriasari, M.Si
Irhamah, M.Si, Ph.D

PROGRAM MAGISTER
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016



TESIS - SS14 2501

MULTI OUTPUT NEURAL NETWORK FOR THE TEMPERATURE FORECASTING IN SEMARANG USING BAYESIAN APPROACH

ZUBDATU ZAHRATI
NRP. 1314 201 037

SUPERVISOR
Dr. Kartika Fithriasari, M.Si
Irhamah, M.Si, Ph.D

MAGISTER PROGRAM
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

MULTI OUTPUT NEURAL NETWORK UNTUK PERAMALAN TEMPERATUR DI SEMARANG DENGAN PENDEKATAN BAYESIAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Sains (M.Si)

Di

Institut Teknologi Sepuluh Nopember

Oleh :

ZUBDATU ZAHIRATI

NRP. 1314 201 037

Tanggal Ujian : 21 Juli 2016

Periode Wisuda : September 2016

Disetujui Oleh :

1. Dr. Kartika Fithriasari, M.Si
NIP. 19691212 199303 2 002

(Pembimbing I)

2. Irhamah, M.Si., Ph.D.
NIP. 19780406 200112 2 002

(Pembimbing II)

3. Prof. Nur Iriawan, M.Ikom., Ph.D.
NIP. 19621015 198803 1 002

(Penguji)

4. Dr. Purnadi, M.Sc
NIP. 19620204 198701 1 001

(Penguji)



Direktur Program Pascasarjana

Prof. Ir. Djauhar Manfaat, M.Sc, Ph.D
NIP 19601202 198701 1 001

MULTI OUTPUT NEURAL NETWORK UNTUK PERAMALAN TEMPERATUR DI SEMARANG DENGAN PENDEKATAN BAYESIAN

Nama mahasiswa : Zubdatu Zahradi
NRP : 1314 201 037
Pembimbing : Dr. Kartika Fithriasari, M.Si.
Co-Pembimbing : Irhamah, M.Si, Ph.D

ABSTRAK

Kota Semarang memiliki pertambahan luasan UHI sebesar rata-rata 8,4% per tahun dalam kurun waktu 1994-2002. Suhu yang tinggi menyebar dengan cepat ke seluruh kota sehingga dapat mengakibatkan dampak negatif. Penelitian mengenai kasus-kasus iklim dan cuaca banyak yang telah menggunakan NN dikarenakan model NN dapat menangkap hubungan nonlinier. Pada penelitian ini akan dilakukan pemodelan temperatur di Semarang dengan menggunakan tiga model yaitu *Vector Autoregressive* (VAR), *Feed Forward Neural Network* (FFNN), dan *Bayesian Neural Network* (BNN). Tujuan dari penelitian ini adalah memperoleh perbandingan hasil kinerja model VAR, FFNN, dan BNN. Penelitian ini akan menggunakan data temperatur di stasiun Semarang dan Ahmad Yani. Data yang digunakan dalam penelitian ini adalah data sekunder dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). Data dibagi menjadi dua tipe, yaitu data *in-sample* dan data *out-of-sample*. Temperatur di stasiun Semarang diduga berkaitan dengan stasiun Ahmad Yani pada waktu yang sama atau berbeda sehingga dilakukan pemodelan secara multivariate dengan model VAR. Pemilihan input optimum berdasarkan pemodelan VAR. Model VAR yang terpilih adalah VARIMA (3,1,0) berdasarkan identifikasi MPACF dan nilai AIC terkecil. Model NN yang digunakan adalah Feed Forward Neural Network (FFNN). Pemilihan model terbaik adalah FFNN (2,2,2) dengan nilai AIC terkecil. . Karena pada model FFNN sering terjadi kasus *overfitting* maka salah satu cara mengatasi adalah *Bayesian Neural Network*. Kriteria *in-sample* menggunakan AIC sedangkan *out-of-sample* menggunakan MAPE dan RMSE. Dari ketiga model tersebut maka dapat diperoleh kesimpulan bahwa model FFNN mempunyai kinerja lebih baik daripada VAR dan BNN berdasarkan kriteria *in-sample*. Sedangkan model BNN mempunyai kinerja lebih baik daripada VAR dan FFNN berdasarkan kriteria *out-of-sample*.

Kata kunci : *Bayesian Neural Network, Feed Forward Neural Network, Temperatur, Vector Autoregressive*

MULTI OUTPUT NEURAL NETWORK FOR THE TEMPERATURE FORECASTING IN SEMARANG USING BAYESIAN APPROACH

Name : Zubdatu Zahradi
NRP : 1314 201 037
Supervisor : Dr. Kartika Fithriasari, M.Si.
Co-Supervisor : Irhamah, M.Si, Ph.D

ABSTRACT

Semarang has an increasing extent of UHI for about 8,4% per year within 1994 sampai dengan 2002. The high temperature that spread rapidly throughout the city causes a negative impact. The research on cases of climate and weather much has been used neural network model because it can capture nonlinear relationship. This study will be done modeling of temperature in Semarang using three models: Vector Autoregressive (VAR), Feed Forward Neural Network (FFNN), and Bayesian Neural Network (BNN). The purpose of this study is to obtain the comparison of the performance of the VAR, FFNN, and BNN model. This study will use temperature data in Semarang and Ahmad Yani station. The data used in this study is secondary data from the Meteorology, Climatology, and Geophysics institution (BMKG). The data were divided into two type, in-sample data and out-of-sample data. The temperature in Semarang station allegedly to be related with Ahmad Yani station at the same or different time each other, so need to be done multivariate modeling with VAR modeling. Selection of the optimal input based on VAR modeling. The selected VAR model is VARIMA (3,1,0) based on MPACF identification and the smallest AIC value. Because of FFNN model often overfitting so one way to overcome is the Bayesian Neural Network. Criteria in-sample using the AIC, while out-of-sample using MAPE and RMSE. Of the three models it can be concluded that the model FFNN has a better performance than the VAR and BNN based on the criteria in-sample. While the model BNN has a better performance than the VAR and FFNN based on the criteria out-of-sample.

Keywords: Bayesian Neural Network, Feed Forward Neural Network, Temperature, Vector Autoregressive

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xi

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	4
1.5 Batasan Penelitian	4

BAB II TINJAUAN PUSTAKA

2.1 Proses Stasioneritas Data <i>Time Series</i>	5
2.2 <i>Matrix Autocorrelation Function</i> (MACF)	6
2.3 <i>Matrix Partial Autocorrelation Function</i> (MPACF)	8
2.4 <i>Vector Autoregressive</i> (VAR)	9
2.5 Uji <i>Nonlinearitas</i>	13
2.6 <i>Neural Network</i> (NN) <i>Multi-Output</i>	14
2.6.1 <i>Backpropagation</i>	16
2.7 Metode Optimasi Levenberg Marquadt	18
2.8 Bayesian <i>Neural Network</i>	20
2.9 Markov Chain Monte Carlo (MCMC)	23
2.10 Kriteria Model Terbaik	25

2.11 Geografi Semarang	26
2.12 Temperatur.....	27
 BAB III METODOLOGI PENELITIAN	
3.1 Sumber Data.....	29
3.2 Metode Analisis	30
 BAB IV ANALISIS DAN PEMBAHASAN	
4.1 Statistika Deskriptif	33
4.2 <i>Vector Autoregressive (VAR)</i>	35
4.3 Uji Nonlinieritas	40
4.4 Pemodelan FFNN	41
4.5 Pemeriksaan Distribusi Data.....	44
4.6 Pemodelan Bayesian <i>Neural Network</i>	45
4.7 Perbandingan Hasil Berdasarkan Kriteria Model Terbaik.....	51
 BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan	53
5.2 Saran	53
 DAFTAR PUSTAKA	55
LAMPIRAN.....	59

DAFTAR TABEL

Tabel 2.1 Transformasi Box-Cox.....	5
Tabel 2.2 Spesifikasi Perbedaan Beberapa Metode Optimasi	18
Tabel 3.1 Struktur Data Penelitian	29
Tabel 4.1 Statistika Deskriptif.....	33
Tabel 4.2 Hasil Uji t masing-masing stasiun sebelum <i>differencing</i>	34
Tabel 4.3 Hasil Uji t masing-masing stasiun setelah <i>differencing</i>	34
Tabel 4.4 Nilai AIC model VAR	37
Tabel 4.5 Estimasi parameter model VARIMA (3,1,0)	37
Tabel 4.6 Nilai AIC residual model VARIMA (3,1,0)	38
Tabel 4.7 Hasil ramalan model VARIMA (3,1,0)	40
Tabel 4.8 Uji Terasvirta	41
Tabel 4.9 Nilai AIC model FFNN	42
Tabel 4.10 Hasil ramalan model FFNN (2,2,2).....	43
Tabel 4.11 Pemeriksaan distribusi data.....	44
Tabel 4.12 Hasil ramalan model FFNN (2,2,2) dengan pendekatan bayesian....	49
Tabel 4.13 Hasil nilai parameter-parameter BNN	50
Tabel 4.14 Hasil nilai parameter-parameter BNN setelah <i>improve</i>	51
Tabel 4.14 Hasil perbandingan model.....	51
Tabel 4.15 Hasil nilai ramalan temperatur harian dengan metode BNN	52

DAFTAR GAMBAR

Gambar 2.1 Contoh Plot MACF	7
Gambar 2.2 Contoh Plot MPACF	8
Gambar 2.3 Arsitektur FFNN <i>multi-output</i>	16
Gambar 4.1 <i>Time series plot</i> temperatur Semarang di stasiun Semarang dan Ahmad Yani	33
Gambar 4.2 <i>Autocorrelation Function</i> (ACF) di stasiun (a) Semarang dan (b) Ahmad Yani sebelum <i>differencing</i>	34
Gambar 4.3 <i>Autocorrelation Function</i> (ACF) di stasiun (a) Semarang dan (b) Ahmad Yani setelah <i>differencing</i>	34
Gambar 4.4 <i>Box-Cox plot</i> temperatur di stasiun (a) Semarang dan (b) Ahmad Yani	35
Gambar 4.5 Plot MACF di stasiun Semarang dan Ahmad Yani sebelum <i>differencing</i>	36
Gambar 4.6 Plot MPACF di stasiun Semarang dan Ahmad Yani sebelum <i>differencing</i>	36
Gambar 4.7 Plot MPACF di stasiun Semarang dan Ahmad Yani setelah <i>differencing</i> 1	36
Gambar 4.8 Hasil <i>q-q plot</i>	39
Gambar 4.9 Hasil <i>timeseries</i> plot ramalan VARIMA (3,1,0)	40
Gambar 4.10 Arsitektur Model FFNN (2,2,2)	42
Gambar 4.11 Hasil <i>timeseries</i> plot ramalan FFNN (2,2,2)	43
Gambar 4.12 <i>Doodle</i> WinBugs FFNN bayesian.....	46
Gambar 4.13 Arsitektur Model Bayesian <i>Neural Network</i>	48
Gambar 4.14 Hasil <i>timeseries</i> plot ramalan BNN antara aktual dan ramalan	49
Gambar 4.15 Hasil perbandingan <i>timeseries plot</i> data aktual dan ramalan	52

DAFTAR LAMPIRAN

Lampiran 1 Data Temperatur Semarang Tahun 2013.....	59
Lampiran 2 Macro SAS untuk VAR.....	60
Lampiran 3 Hasil Estimasi Parameter Model VARIMA (3,1,0)	61
Lampiran 3.1 Sebelum <i>Restrict</i>	61
Lampiran 3.2 Setelah <i>Restrict</i>	61
Lampiran 4 Uji Multivariat Normal.....	62
Lampiran 5 Uji Terasvirta Menggunakan Program R	63
Lampiran 6 Mfile Matlab FFNN (2,2,2)	65
Lampiran 7 Pemeriksaan Distribusi Data	67
Lampiran 7.1 Pemeriksaan Distribusi Normal Data Temperatur di Stasiun Semarang	67
Lampiran 7.2 Pemeriksaan Distribusi Normal Data Temperatur di Stasiun Ahmad Yani.....	67
Lampiran 8 Syntax WinBugs Bayesian Neural Network	68
Lampiran 9 ACF Bayesian Neural Network.....	70
Lampiran 10 Time series plot parameter-parameter BNN.....	72
Lampiran 11 ACF Bayesian Neural Network setelah <i>improve</i>	75
Lampiran 12 Time series plot parameter-parameter BNN setelah <i>improve</i>	77

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kota Semarang merupakan salah satu kota besar di Indonesia yang mengalami kenaikan suhu yang cukup tinggi sejak tahun 1990 yang dapat dilihat dari meningkatnya luasan kawasan UHI (LIPI, 2014). Kota Semarang memiliki pertambahan luasan UHI sebesar rata-rata 8,4% per tahun dalam kurun waktu 1994-2002. Suhu yang tinggi menyebar dengan cepat ke seluruh kota karena wilayah terbangun di perkotaan juga terus menyebar di seluruh bagian wilayah perkotaan (Tursilowati, 2007). Suhu udara lebih besar dari 34°C pada tahun 1994 di Kota Semarang hanya berada pada beberapa titik, namun pada tahun 2002 sudah hampir menutupi seluruh Kota Semarang. Sehingga dapat meningkatkan suhu permukaan global yang dapat memicu terjadinya perubahan iklim.

Peningkatan suhu rata-rata di atmosfer, laut, dan daratan dapat menyebabkan terjadi pemanasan global. Sejak pertengahan abad ke-20, terjadi peningkatan sebesar $0,74 \pm 0,18^{\circ}\text{C}$ suhu udara rata-rata global pada permukaan bumi. Kemungkinan besar terjadi karena meningkatnya konsentrasi gas-gas rumah kaca (IPCC, 2007). Selain itu, dapat menimbulkan kekeringan dan naiknya pemakaian pendingin ruangan yang secara tidak langsung mengakibatkan meningkatnya pemakaian listrik (Kvalevag, 2009). Kelembaban udara dan temperatur dapat mempengaruhi kondisi fisik manusia yang dapat menyebabkan kelelahan, menurunkan daya konsentrasi bahkan dapat menurunkan kekuatan-kekuatan otot yang berakibat menurunkan produktivitas (Purnomo & Rizal, 2000). Untuk mengatasi masalah tersebut, salah satu upaya yang dapat dilakukan adalah meramalkan temperatur untuk beberapa periode ke depan dengan menggunakan *Neural Network* (NN) pendekatan Bayesian.

Penelitian mengenai kasus-kasus iklim dan cuaca banyak yang telah menggunakan NN dikarenakan model NN dapat menangkap hubungan *nonlinier*. Peramalan temperatur pernah dilakukan di kota Surabaya dengan membandingkan antara metode ARIMA dan *Neural Network* (NN) yang diperoleh kesimpulan

bahwa ramalan NN lebih baik daripada ARIMA (Machmudin & Ulama, 2012). Penelitian dengan NN pendekatan Bayesian untuk memprediksi curah hujan per jam diperoleh hasil bahwa Bayesian *Neural Network* (BNN) *with adjusting* lebih baik daripada *natural* BNN dan FFNN (Fithriasari, Iriawan, Ulama, Sutikno, dan Kuswanto, 2013). Selain itu, penelitian selanjutnya juga menghasilkan kesimpulan bahwa model BNN lebih baik daripada FFNN (Fithriasari, Iriawan, Ulama, dan Sutikno, 2013). Penelitian mengenai temperatur telah dilakukan di kota Surabaya yang menghasilkan kesimpulan bahwa metode BNN lebih baik untuk *in sample* sedangkan untuk kriteria *out sample* lebih baik menggunakan *Neural Network* (Permai, 2013).

Neural Network merupakan salah satu contoh model *nonlinear* yang mempunyai bentuk fungsional fleksibel dan mengandung beberapa parameter yang tidak dapat diinterpretasikan seperti pada model parametrik. Salah satu bentuk model NN yang banyak digunakan untuk berbagai aplikasi adalah *Feed Forward Neural Network* (FFNN). Peramalan data *time series* telah banyak digunakan untuk aplikasi model NN. Dalam penerapannya, FFNN mengandung sejumlah parameter (*weight*) yang terbatas. Untuk mendapatkan model FFNN yang sesuai dengan menentukan kombinasi yang tepat antara banyaknya variabel dan banyaknya unit pada lapis tersembunyi. Hal ini berdampak pada banyaknya parameter yang optimal (Suhartono, 2007).

Kelemahan dari metode *neural network* adalah sering mengalami *overfitting* yaitu pada data *in-sample* hasil peramalan baik tetapi kurang baik pada data *out-of-sample*. Ada beberapa cara untuk mengatasi hal ini yaitu dengan melakukan *early learning stopping* dan pendekatan Bayesian (Doan & Liong, 2004). *Bayesian learning* pada NN digunakan untuk mendapatkan bobot parameter-parameter yang cocok untuk data training namun mencegah terjadinya *overfitting*. Pendekatan Bayesian menghasilkan keakuratan prediksi yang lebih tinggi daripada *early learning stopping* (Neal, 1995). Metode Bayesian berbeda dengan metode *frequentist* yang hanya mempertimbangkan informasi dari data. Metode Bayesian menggabungkan informasi dari data dengan informasi masa lalu yang disebut dengan distribusi prior. Oleh karena itu, hasil dari Bayesian bergantung pada penggunaan distribusi prior yang tepat.

Penelitian ini menggunakan data temperatur kota Semarang di stasiun Semarang dan Ahmad Yani. Temperatur pada satu stasiun diduga berkaitan dengan stasiun lainnya pada waktu yang sama atau berbeda, sehingga pada penelitian ini akan dilakukan pemodelan secara *multivariate*. Pemodelan dan peramalan temperatur kota Semarang menggunakan *Neural Network* (NN) dengan pendekatan Bayesian (BNN). Input yang digunakan pada model *Feed Forward Neural Network* (FFNN) adalah hasil dari pemodelan VAR. Karena pada metode *Neural Network* sering terjadi kasus *overfitting* maka akan dilanjutkan dengan metode Bayesian *Neural Network*. Sehingga dapat diperoleh hasil perbandingan model *Neural Network* (NN) dan Bayesian *Neural Network* (BNN) berdasarkan hasil kinerja model untuk mendapatkan nilai ramalan yang akurat.

1.2 Perumusan Masalah

Penelitian ini menggunakan dua stasiun yaitu stasiun Semarang dan Ahmad Yani yang diduga saling berkaitan pada waktu yang sama atau berbeda. Oleh karena itu, analisis dilakukan secara *multivariate* dengan pemodelan VAR. Berdasarkan latar belakang yang telah diuraikan maka permasalahan yang akan dibahas dalam penelitian ini adalah bagaimana bentuk model *Neural Network* dan Bayesian *Neural Network* yang sesuai untuk kasus temperatur harian di Semarang. Selanjutnya, bagaimana perbandingan hasil kinerja model *Neural Network* dan Bayesian *Neural Network* dalam meramalkan temperatur di Semarang.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penelitian yang ingin dicapai dalam penelitian ini adalah sebagai berikut :

1. Memperoleh variabel *input Neural Network* secara *multivariate* dari hasil pemodelan VAR .
2. Memperoleh model *Neural Network* dan Bayesian *Neural Network* yang sesuai untuk peramalan kasus temperatur harian di Semarang.
3. Memperoleh perbandingan hasil kinerja model *Neural Network* dan Bayesian *Neural Network*.

1.4 Manfaat Penelitian

Manfaat yang ingin diperoleh dari hasil penelitian ini adalah sebagai berikut:

1. Bagi peneliti, memberikan wawasan keilmuan yang berhubungan dengan pengembangan dan penerapan *Neural Network* dengan pendekatan Bayesian.
2. Bagi BMKG, memberikan masukan dalam memperoleh model temperatur di kota Semarang.

1.5 Batasan Penelitian

Batasan penelitian pada permasalahan ini adalah stasiun cuaca yang digunakan di Semarang yaitu stasiun Semarang dan Ahmad Yani dengan mempertimbangkan pengaruh antar stasiun dan tanpa mempertimbangkan variabel lain yang mempengaruhi temperatur. Fungsi aktivasi dari lapisan masukan ke lapisan tersembunyi menggunakan logistik sigmoid. Sedangkan dari lapisan tersembunyi ke lapisan keluaran menggunakan fungsi aktivasi linier.

BAB II

TINJAUAN PUSTAKA

2.1 Proses Stasioneritas Data *Time Series*

Deret waktu adalah suatu deret (barisan) dari pengamatan Y_t pada suatu variabel Y , dimana tiap-tiap pengamatan dicatat pada suatu waktu tertentu $t \in T$. Dalam hal ini T adalah himpunan dari waktu dimana pengamatan-pengamatan tersebut dilakukan. Sifat penting untuk analisis deret waktu adalah proses-proses (stokastik) tersebut merupakan proses yang stasioner, yaitu independen terhadap waktu (Suhartono, 2007). Kondisi stasioner terdiri atas dua hal, yaitu stasioner dalam varians dan stasioner dalam rata-rata (Wei, 2006). Pendeteksian stasioner dalam varians menggunakan transformasi Box-Cox sedangkan pengujian stasioner dalam rata-rata menggunakan uji *Augmented Dickey-Fuller* (ADF). Nilai estimasi λ dapat dilihat dari Tabel 2.1 atau menggunakan rumus sebagai berikut.

$$T(Z_t) = Z_t^{(\lambda)} = \frac{Z_t^\lambda - 1}{\lambda} \quad (2.1)$$

Tabel 2.1 Transformasi Box-Cox

Nilai estimasi λ	Transformasi
-1,0	$1/Y_t$
-0,5	$1/\sqrt{Y_t}$
0	$\ln Y_t$
0,5	$\sqrt{Y_t}$
1	Y_t (tidak ada transformasi)

Langkah selanjutnya adalah menguji stasioneritas data dalam rata-rata. Data dikatakan stasioner dalam rata-rata jika berfluktuasi di sekitar suatu nilai rata-rata yang konstan maka dikatakan deret stasioner dalam rata-rata. Selain itu, dapat dilihat dari plot fungsi autokorelasi dapat digunakan juga sebagai alat untuk mengidentifikasi kestasioneran data, jika plot cenderung turun lambat maka dapat disimpulkan data belum stasioner dalam rata-rata. Selain itu, dapat juga diuji dengan menggunakan *Augmented Dickey-Fuller* untuk mengetahui data telah

stasioner dalam rata-rata atau belum dengan persamaan regresi sebagai berikut (Gujarati, 2004).

1. Persamaan regresi merupakan model *random walk*

$$\Delta Y_t = \delta Y_{t-1} + \sum_{i=1}^p \alpha_i \Delta Y_{t-i} + a_t \quad (2.2)$$

2. Persamaan regresi merupakan model dengan penambahan intersep

$$\Delta Y_t = \beta_1 + \delta Y_{t-1} + \sum_{i=1}^p \alpha_i \Delta Y_{t-i} + a_t \quad (2.3)$$

3. Persamaan regresi merupakan model dengan penambahan intersep dan *trend* stokastik

$$\Delta Y_t = \beta_1 + \beta_2 t + \delta Y_{t-1} + \sum_{i=1}^p \alpha_i \Delta Y_{t-i} + a_t \quad (2.4)$$

Pengujian ADF menggunakan hipotesis sebagai berikut.

$H_0 : \delta = 0$ (Data tidak stasioner)

$H_1 : \delta < 0$ (Data stasioner)

Statistik uji yang digunakan adalah

$$t = \frac{\hat{\delta} - 1}{SE(\hat{\delta})} \quad (2.5)$$

$$SE(\hat{\delta}) = \sqrt{\frac{n \hat{\sigma}_a^2}{n \sum_{t=1}^n Y_{t-1}^2 - \left(\sum_{t=1}^n Z_{t-1} \right)^2}} \quad , \quad \hat{\sigma}_a^2 = \sum_{t=1}^n \frac{(Y_t - \hat{\beta} - \hat{\delta} Y_{t-1})^2}{(n-2)} \quad .$$

Daerah kritis untuk pengujian *Augmented Dickey-Fuller* adalah H_0 akan ditolak jika $|t| > t_{\alpha, n-1}$ dengan taraf signifikan sebesar α .

2.2 Matriks Autocorrelation Function (MACF)

Apabila terdapat vektor deret waktu dengan n observasi Y_1, Y_2, \dots, Y_n maka dapat dihitung matriks korelasi sampelnya sebagai berikut (Wei, 2006).

$$\hat{\rho}(k) = \left[\hat{\rho}_{ij}(k) \quad , \quad i = 1, 2, \dots, p \quad j = 1, 2, \dots, m \right] \quad (2.6)$$

Dimana $\hat{\rho}_{ij}(k)$ adalah korelasi silang antara komponen *series* ke-*i* dengan ke-*j* yang dinyatakan dalam persamaan sebagai berikut.

$$\hat{\rho}_{ij}(k) = \frac{\sum_{t=1}^{n-k} (Y_{i,t} - \bar{Y}_i)(Y_{j,t+k} - \bar{Y}_j)}{\left[\sum_{t=1}^n (Y_{i,t} - \bar{Y}_i)^2 \sum_{t=1}^n (Y_{j,t} - \bar{Y}_j)^2 \right]^{1/2}} = \frac{\hat{\gamma}_k}{\hat{\gamma}_0} \quad (2.7)$$

dengan \bar{Y}_i dan \bar{Y}_j merupakan rata-rata sampel dari komponen *series* yang saling bersesuaian, $\hat{\gamma}_k$ adalah penaksir kovarian pada lag ke-*k*, dan $\hat{\gamma}_0$ adalah penaksir varians.

Persamaan matriks korelasi sampel ini sangat berguna untuk menentukan orde dalam *moving average* (MA) yaitu lag *q* pada proses MA(*q*).

Schematic Representation of Cross Correlations																			
Variabel/ Lag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
y1	++	+.	-.	..	--	.-	+.-	..	+.-
y2	++	-.	--	.-	.--	+.	.---
+ is > 2*std error, - is < -2*std error, . is between																			

Gambar 2.1 Contoh Plot MACF

Sebuah metode koefisien diperkenalkan dengan tujuan untuk meringkas sampel korelasi (Box & Tiao, 1992). Metode ini menggunakan symbol (+), (-), dan (.) pada baris ke-*i* dan kolom ke-*j* pada matrik sampel korelasi, dimana telah ditunjukkan pada Gambar 2.1.

- 1) Simbol (+) menunjukkan bahwa nilai $\hat{\rho}_{ij}(k)$ lebih besar dari 2 kali nilai estimasi *standard error* (SE), yang artinya bahwa komponen (*i*, *j*) memiliki korelasi positif.
- 2) Simbol (-) menunjukkan bahwa nilai $\hat{\rho}_{ij}(k)$ lebih kecil dari -2 kali nilai estimasi *standard error* (SE), yang artinya bahwa komponen (*i*, *j*) memiliki korelasi negatif.
- 3) Simbol (.) menunjukkan bahwa nilai $\hat{\rho}_{ij}(k)$ yang terletak diantara -2 sampai 2 dari nilai estimasi *standard error* (SE), artinya bahwa komponen (*i*, *j*) tidak memiliki korelasi.

2.3 Matriks Partial Autocorrelation Function (MPACF)

Partial Autocorrelation Function dalam *univariate time series* digunakan untuk mengidentifikasi orde dalam model AR(p), dimana $\Phi_{kk} = 0$ untuk $|k| > p$, begitu pula dengan matriks korelasi parsial secara *multivariate*. *Partial lag autocorrelation matrix* digunakan untuk mengidentifikasi model vektor AR(p).

$$\mathbf{u}_{s-1,t+s} = \mathbf{Y}_{t+s} - \mathbf{a}_{s-1,1}\mathbf{Y}_{t+s-1} - \dots - \mathbf{a}_{s-1,s-1}\mathbf{Y}_{t+1} = \begin{cases} \mathbf{Y}_{t+s} - \sum_{k=1}^{s-1} \mathbf{a}_{s-1,k} \mathbf{Y}_{t+s-k}, & s \geq 2 \\ \mathbf{Y}_{t+1}, & s = 1 \end{cases} \quad (2.8)$$

dan

$$\mathbf{v}_{s-1,t} = \mathbf{Y}_t - \mathbf{\beta}_{s-1,1}\mathbf{Y}_{t+1} - \dots - \mathbf{\beta}_{s-1,s-1}\mathbf{Y}_{t+s-1} = \begin{cases} \mathbf{Y}_t - \sum_{k=1}^{s-1} \mathbf{\beta}_{s-1,k} \mathbf{Y}_{t+k}, & s \geq 2 \\ \mathbf{Y}_t, & s = 1 \end{cases} \quad (2.9)$$

Kovarian antara $\mathbf{v}_{s-1,t}$ dan $\mathbf{u}_{s-1,t+s}$ adalah

$$\begin{aligned} \text{Cov}(\mathbf{v}_{s-1,t}, \mathbf{u}_{s-1,t+s}) &= E[(Y_t - \beta(s)Y_t(s))(Y_{t+s} - \alpha(s)Y_t(s))'] \\ &= \Gamma(s) - b'(s)[A(s)]^{-1}c(s) = \Gamma(s) - b'(s)\alpha'(s) \end{aligned} \quad (2.10)$$

Sehingga *Partial lag autocorrelation matrix* pada lag s didefinisikan sebagai berikut.

$$\mathbf{P}(s) = [\mathbf{D}_v(s)]^{-1} \mathbf{V}_{vu}(s) [\mathbf{D}_u(s)]^{-1} \quad (2.11)$$

dengan $\mathbf{D}_v(s)$ adalah diagonal matrik dimana elemen diagonal ke- i adalah akar elemen diagonal ke- i dari $\mathbf{V}_v(s)$ dimana $\mathbf{V}_v(s)$ adalah $\text{var}(\mathbf{v}_{s-1,t})$, dan $\mathbf{V}_{vu}(s)$ adalah $\text{cov}(\mathbf{v}_{s-1,t}, \mathbf{u}_{s-1,t+s})$. $\mathbf{D}_u(s)$ didefinisikan sama dengan $\mathbf{V}_u(s)$ dimana $\mathbf{V}_u(s)$ adalah $\text{var}(\mathbf{u}_{s-1,t+s})$. Elemen $\hat{\mathbf{P}}(s)$ dinotasikan dengan $\hat{P}_{ij}(s)$. Berikut adalah contoh pola MPACF pada output *software* SAS yang ditunjukkan pada Gambar 2.2.

Schematic Representation of Partial Cross Correlations																		
Variabel/Lag	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
y1
y2
+ is > 2*std error, - is < -2*std error, . is between																		

Gambar 2.2 Contoh Plot MPACF

Tanda positif (+) menotasikan bahwa nilai $\hat{P}_{ij}(s)$ lebih besar dari $\frac{2}{\sqrt{n}}$ sehingga menunjukkan adanya hubungan korelasi positif. Sedangkan tanda negatif (-) menotasikan bahwa nilai $\hat{P}_{ij}(s)$ lebih kecil dari $-\frac{2}{\sqrt{n}}$ sehingga bernilai negatif menunjukkan adanya hubungan korelasi negatif, dan untuk tanda titik (.) menotasikan nilainya berada antara $\frac{2}{\sqrt{n}}$ dan $-\frac{2}{\sqrt{n}}$, menunjukkan bahwa tidak ada hubungan atau korelasi.

2.4 Vector Autoregressive (VAR)

Vector Autoregressive (VAR) merupakan salah satu pemodelan linier dalam analisis *multivariate time series*. Sebelum dilakukan pemodelan VAR, terlebih dahulu dilakukan pemeriksaan stasioneritas dalam varians dan mean pada data. Apabila data tidak stasioner dalam varians maka dilakukan transformasi Box-Cox. Sedangkan bila data tidak stasioner dalam mean maka dilakukan *differencing*. Secara umum persamaan model VAR(p) dapat ditulis seperti berikut (Wei, 2006).

$$\mathbf{Y}_t = \Phi_1 \mathbf{Y}_{t-1} + \dots + \Phi_p \mathbf{Y}_{t-p} + \mathbf{a}_t \quad (2.12)$$

Bentuk matriks adalah sebagai berikut.

$$\begin{bmatrix} Y_{1t} \\ Y_{2t} \\ \vdots \\ Y_{mt} \end{bmatrix} = \begin{bmatrix} \phi_{111} & \phi_{112} & \dots & \phi_{11m} \\ \phi_{121} & \phi_{122} & \dots & \phi_{12m} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{1m1} & \phi_{1m2} & \dots & \phi_{1mm} \end{bmatrix} \begin{bmatrix} Y_{1(t-1)} \\ Y_{2(t-1)} \\ \vdots \\ Y_{m(t-1)} \end{bmatrix} + \dots + \begin{bmatrix} \phi_{p11} & \phi_{p12} & \dots & \phi_{p1m} \\ \phi_{p21} & \phi_{p22} & \dots & \phi_{p2m} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{pm1} & \phi_{pm2} & \dots & \phi_{pmm} \end{bmatrix} \begin{bmatrix} Y_{1(t-p)} \\ Y_{2(t-p)} \\ \vdots \\ Y_{m(t-p)} \end{bmatrix} + \begin{bmatrix} a_{1t} \\ a_{2t} \\ \vdots \\ a_{mt} \end{bmatrix}$$

Keterangan:

- m : banyak stasiun, dimana stasiun yang digunakan adalah Semarang dan Ahmad Yani
- Y_t : vektor variabel Y berukuran $m \times 1$ pada waktu ke- t
- Y_{t-1} : vektor variabel Y berukuran $m \times 1$ pada waktu ke- $(t-1)$
- Φ_1 : matriks parameter variabel Y orde 1 (waktu ke- t) berukuran $m \times m$
- Φ_p : matriks parameter variabel Y orde p (waktu ke- t) berukuran $m \times m$

a_t : vektor residual berukuran $m \times 1$ pada waktu ke- t

Data *time series* dengan banyak data sebanyak n untuk masing-masing variabel sebanyak m maka kita dapat mendefinisikan beberapa notasi sebagai berikut (Lutkepohl, 2005).

$$\mathbf{Y} = (y_1, y_2, \dots, y_n)_{(m \times n)},$$

$$\mathbf{B} = (\boldsymbol{\mu}, \boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \dots, \boldsymbol{\Phi}_p)_{(m \times (mp+1))},$$

$$\mathbf{X}_t = \begin{pmatrix} 1 \\ \mathbf{y}_t \\ \vdots \\ \mathbf{y}_{t-p+1} \end{pmatrix}_{((mp+1) \times 1)},$$

$$\mathbf{X} = (\mathbf{X}_0, \dots, \mathbf{X}_{n-1})_{((mp+1) \times n)},$$

$$\mathbf{E} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)_{(m \times n)},$$

$$\mathbf{y} = (\text{vec}(\mathbf{Y}))_{(mn \times 1)},$$

$$\mathbf{Q} = (\text{vec}(\mathbf{B}))_{((m^2 p + m) \times 1)},$$

$$\mathbf{q} = (\text{vec}(\mathbf{B}^T))_{((m^2 p + m) \times 1)},$$

$$\mathbf{e} = (\text{vec}(\mathbf{E}))_{(mn \times 1)},$$

dengan menggunakan notasi-notasi tersebut untuk $t = 1, 2, \dots, n$, maka model VAR(p) pada persamaan (2.12) dapat dituliskan sebagai berikut.

$$\mathbf{Y} = \mathbf{B}\mathbf{X} + \mathbf{E},$$

$$\text{vec}(\mathbf{Y}) = (\mathbf{X}^T \otimes \mathbf{I}_m) \text{vec}(\mathbf{B}) + \text{vec}(\mathbf{E}),$$

$$\mathbf{y} = (\mathbf{X}^T \otimes \mathbf{I}_m) \mathbf{Q} + \mathbf{e} \quad (2.13)$$

Matriks kovarians dari \mathbf{e} adalah

$$\boldsymbol{\Sigma}_e = \mathbf{I}_n \otimes \boldsymbol{\Sigma}_e, \quad (2.14)$$

Maka estimasi *Least Square* (LS) multivariat atau *Generalized Least Square* (GLS) dari \mathbf{Q} berarti memilih estimator yang meminimalkan $S(\mathbf{Q})$ dengan rumus sebagai berikut.

$$\begin{aligned}
S(\mathbf{Q}) &= \mathbf{e}^T (\mathbf{I}_n \otimes \Sigma_e)^{-1} \mathbf{e} \\
&= \mathbf{e}^T (\mathbf{I}_n \otimes \Sigma_e^{-1}) \mathbf{e} \\
S(\mathbf{Q}) &= (\mathbf{y} - (\mathbf{X}^T \otimes \mathbf{I}_m) \mathbf{Q})^T (\mathbf{I}_n \otimes \Sigma_e^{-1}) (\mathbf{y} - (\mathbf{X}^T \otimes \mathbf{I}_m) \mathbf{Q}) \\
&= \text{vec}(\mathbf{Y} - \mathbf{B}\mathbf{X})^T (\mathbf{I}_n \otimes \Sigma_e^{-1}) \text{vec}(\mathbf{Y} - \mathbf{B}\mathbf{X}) \\
&= \text{tr} \left[(\mathbf{Y} - \mathbf{B}\mathbf{X})^T \Sigma_e^{-1} (\mathbf{Y} - \mathbf{B}\mathbf{X}) \right]
\end{aligned} \tag{2.15}$$

dengan tujuan untuk meminumkan persamaan (2.15), maka:

$$\begin{aligned}
S(\mathbf{Q}) &= \mathbf{y}^T (\mathbf{I}_n \otimes \Sigma_e)^{-1} \mathbf{y} + \mathbf{Q}^T (\mathbf{X} \otimes \mathbf{I}_m) (\mathbf{I}_n \otimes \Sigma_e^{-1}) (\mathbf{X}^T \otimes \mathbf{I}_m) \mathbf{Q} \\
&\quad - 2\mathbf{Q}^T (\mathbf{X} \otimes \mathbf{I}_m) (\mathbf{I}_n \otimes \Sigma_e^{-1}) \mathbf{y} \\
&= \mathbf{y}^T (\mathbf{I}_n \otimes \Sigma_e)^{-1} \mathbf{y} + \mathbf{Q}^T (\mathbf{X}\mathbf{X}^T \otimes \Sigma_e^{-1}) \mathbf{Q} - 2\mathbf{Q}^T (\mathbf{X} \otimes \Sigma_e^{-1}) \mathbf{y}
\end{aligned} \tag{2.16}$$

Selanjutnya persamaan (2.16) diturunkan terhadap \mathbf{Q} sehingga diperoleh hasil:

$$\frac{\partial S(\mathbf{Q})}{\partial \mathbf{Q}} = 2(\mathbf{X}\mathbf{X}^T \otimes \Sigma_e^{-1}) \mathbf{Q} - 2(\mathbf{X} \otimes \Sigma_e^{-1}) \mathbf{y} \tag{2.17}$$

Persamaan (2.17) disamadengankan nol, sehingga didapatkan persamaan normal yaitu:

$$\begin{aligned}
(\mathbf{X}\mathbf{X}^T \otimes \Sigma_e^{-1}) \hat{\mathbf{Q}} &= (\mathbf{X} \otimes \Sigma_e^{-1}) \mathbf{y} \\
\hat{\mathbf{Q}} &= \left((\mathbf{X}\mathbf{X}^T)^{-1} \otimes \Sigma_e \right) (\mathbf{X} \otimes \Sigma_e^{-1}) \mathbf{y} \\
\hat{\mathbf{Q}} &= \left((\mathbf{X}\mathbf{X}^T)^{-1} \mathbf{X} \otimes \mathbf{I}_m \right) \mathbf{y}
\end{aligned} \tag{2.18}$$

Hessian dari $S(\mathbf{Q})$ adalah

$$\frac{\partial^2 S(\mathbf{Q})}{\partial \mathbf{Q} \partial \mathbf{Q}^T} = 2(\mathbf{X}\mathbf{X}^T \otimes \Sigma_e^{-1}) .$$

Hessian tersebut merupakan matriks definit positif yang mengkonfirmasi bahwa $\hat{\mathbf{Q}}$ benar-benar meminimalkan vektor.

Setelah mendapatkan estimasi model VAR maka langkah selanjutnya adalah menguji signifikansi parameter dengan statistik uji t . Hipotesis yang digunakan dalam pengujian model VAR adalah sebagai berikut.

$$H_0 : \phi_{ijk} = 0$$

$$H_1 : \phi_{ijk} \neq 0 \quad , i = 1, 2, \dots, p; j = 1, 2, \dots, m; k = 1, 2, \dots, m$$

Statistik uji yang digunakan untuk pengujian model VAR adalah sebagai berikut.

$$t = \frac{\hat{\phi}_{ijk}}{SE(\hat{\phi}_{ijk})} \quad (2.19)$$

Daerah kritis untuk pengujian model VAR adalah H_0 akan ditolak jika $|t| > t_{\alpha/2; (n-p-1)}$ dimana n yaitu banyaknya data dan p merupakan banyaknya parameter yang signifikan dengan taraf signifikan sebesar α . $SE(\hat{\phi}_{ijk})$ merupakan nilai *standard error* yang didapatkan melalui akar dari elemen diagonal ke- i dari $(\mathbf{XX}^T)^{-1} \otimes \hat{\Sigma}_e$, dimana:

$$\hat{\Sigma}_e = \frac{1}{n} \sum_{t=1}^n \hat{\mathbf{e}}_t \hat{\mathbf{e}}_t' \quad ; \quad \mathbf{X} = \begin{pmatrix} 1 & Y_{t-1} & \dots & Y_{t-p} \end{pmatrix}^T .$$

Ada beberapa asumsi yang harus dipenuhi pada model VAR yaitu residual *white noise* dan multivariate normal. Uji *white noise* digunakan untuk menguji signifikansi secara keseluruhan pada autokorelasi residual hingga lag h . Pengujian dilakukan dengan cara memodelkan ulang residual dari model. Selanjutnya dilakukan pengecekan dimana letak AIC terkecil. Jika letak AIC terkecil terdapat pada lag AR(0) dan MA(0), maka residual dapat dikatakan sudah memenuhi asumsi *white noise*. Selanjutnya dilakukan pemeriksaan asumsi bahwa vektor residual berdistribusi multivariat normal dengan cara membuat q-q plot. Tahapan dari pembuatan q-q plot adalah sebagai berikut (Johnson & Winchern, 2007).

1. Menghitung nilai vektor rata-rata : \bar{X}
2. Menghitung nilai matriks varians-kovarians : S
3. Menghitung nilai jarak Mahalanobis setiap titik pengamatan dengan vektor rata-ratanya $d_j^2 = \left((X_j - \bar{X})' S^{-1} (X_j - \bar{X}) \right)$, $j = 1, \dots, n$
4. Mengurutkan nilai d_j^2 dari kecil ke besar : $d_{(1)}^2 \leq d_{(2)}^2 \leq d_{(3)}^2 \leq \dots \leq d_{(n)}^2$

5. Menghitung nilai $p_j = \frac{j - \frac{1}{2}}{n}$, $j = 1, \dots, n$
6. Menghitung nilai q_j sedemikian hingga $\int_{-\infty}^{q_j} f(\chi^2) d\chi^2 = p_j$
7. Membuat *scatterplot* antara d_j^2 dengan q_j
8. Jika *scatter-plot* ini cenderung membentuk garis lurus dan lebih dari 50 % nilai $d_j^2 \leq \chi_{p,0.50}^2$ maka dapat dikatakan bahwa residual telah berdistribusi normal multivariat.

2.5 Uji *Nonlinearitas*

Tahap pertama yang harus dilakukan adalah menguji hubungan linieritas pada data deret waktu. Secara visual, hubungan tidak linear dapat dilihat dengan plot data pada lag-lag data. Selain itu, jika secara inferensi dapat dilakukan dengan pengujian. Metode yang digunakan pada penelitian ini untuk menguji *nonlinearitas* adalah uji Terasvirta.

Uji Terasvirta merupakan salah satu langkah uji yang digunakan untuk mendeteksi hubungan tidak linear. Uji Terasvirta merupakan pengembangan dari model *Neural Network* yang dikemukakan oleh Terasvirta (Terasvirta, Lin, & Granger, 1993). Model *Neural Network* dengan satu lapisan tersembunyi diberikan sebagai berikut.

$$y_t = \beta' w_t + \sum_{j=1}^q \theta_{0j} \left\{ \psi \left(\gamma_j' w_t \right) - \frac{1}{2} \right\} + \varepsilon_t \quad (2.20)$$

Hipotesis yang digunakan untuk uji Terasvirta adalah sebagai berikut.

$$H_0 : \theta_{0j} = 0 \quad (\text{model Linier})$$

$$H_1 : \theta_{0j} \neq 0 \quad (\text{model Tidak Linier})$$

Kandidat yang natural dari kinerja suatu *network* adalah kuadrat dari koefisien korelasi antara y dan \hat{y} sebagai berikut.

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.21)$$

dimana \hat{y} adalah vektor dari *output network* dan R^2 adalah koefisien determinasi regresi linier dari *error* (Lee, White, & Granger, 1993). Statistik uji yang digunakan pada uji Terasvirta adalah sebagai berikut.

$$\chi^2 = nR^2 \quad (2.22)$$

Daerah kritis yang digunakan adalah H_0 dan akan ditolak jika nilai $\chi^2 > \chi_{\alpha; p}^2$ dimana p adalah banyaknya variabel prediktor dan n adalah banyaknya sampel. Prosedur pengujian dengan langkah-langkah sebagai berikut.

1. Meregresikan y_t dengan variabel prediktor, kemudian menghitung nilai residual

$$\hat{\varepsilon} = y_t - \hat{y}_t.$$

2. Meregresikan residual \hat{u} dengan variabel prediktor dan kemudian hitung koefisien determinasi dari regresi (R^2).

3. Menghitung nilai statistik uji $\chi^2 = nR^2$ dimana nilai tersebut dibandingkan dengan χ_p^2 , dimana p adalah banyaknya variabel prediktor dan n adalah banyaknya sampel. Hipotesis H_0 ditolak jika nilai statistik uji melebihi nilai dari χ_p^2 .

2.6 Neural Network (NN) Multi-Output

Neural Network (NN) adalah perangkat pengolahan data tidak linear yang dibangun dari perangkat dasar yang saling terhubung disebut dengan neuron dan terdapat bobot tiap koneksi. NN diaplikasikan untuk analisis identifikasi pola data atau klasifikasi data (Sivanandam, Sumathi, & Deepa, 2006). Pada lapisan tersembunyi, fungsi aktivasi yang sering digunakan dalam NN adalah fungsi *logistic sigmoid* sebagai berikut (Bishop, 1995).

$$f_1(a) = \frac{1}{1 + \exp(-a)} \quad (2.23)$$

FFNN mempunyai *node* lebih dari satu pada lapisan keluaran untuk kasus deret waktu *multi-output*. Secara umum struktur FFNN deret waktu dengan K -keluaran dijelaskan pada Gambar 2.3 dimana jika terdapat K variabel random $Y_{1(t)}, Y_{2(t)}, \dots, Y_{K(t)}$ dan d -lag time untuk deret waktu (Fithriasari, 2014). Arsitektur pada Gambar 2.3 mempunyai *node* pada lapisan masukan sebanyak $I = K \times d$ dan *node* pada lapisan keluaran sebanyak K .

Struktur *Neural Network* terdiri dari lapisan masukan, tersembunyi dan keluaran dimana disebut FFNN (I, J, K) atau dapat ditulis dalam persamaan 2.24. Variabel keluaran pada model FFNN adalah $Y_{1(t)}, Y_{2(t)}, \dots, Y_{K(t)}$ dan variabel masukan adalah $X_{1(t)} = Y_{1(t-1)}, \dots, X_{K(t)} = Y_{K(t-1)}, X_{K+1(t)} = Y_{1(t-2)}, \dots, X_{I(t)} = Y_{K(t-d)}$.

$$Y_{k(t)} = V_{0k} + \sum_{j=1}^J V_{jk} f_1 \left(W_{0j} + \sum_{i=1}^I X_{i(t)} W_{ij} \right) + \varepsilon_{k(t)}, \quad k = 1, \dots, K \quad t = d + 1, \dots, n \quad (2.24)$$

dimana:

W_{ij} : bobot *node* i pada lapisan masukan ke *node* j pada lapisan tersembunyi

W_{0j} : bias untuk *node* j pada lapisan tersembunyi

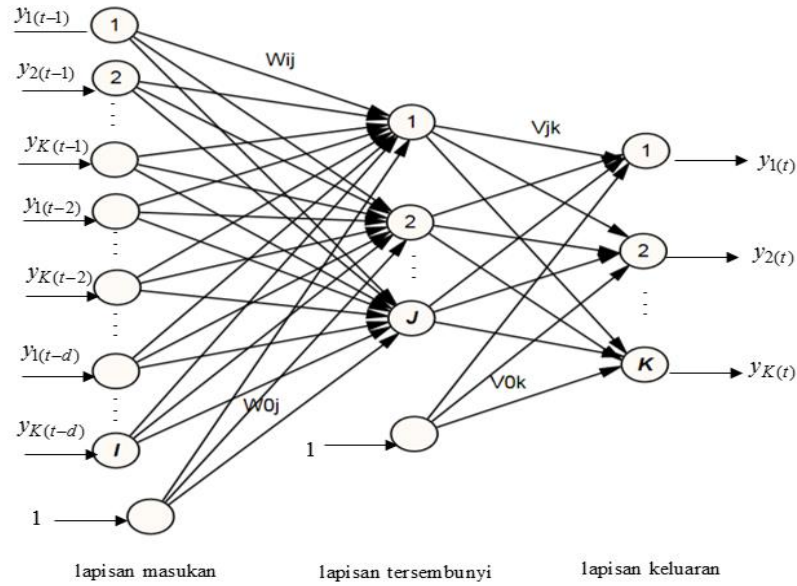
V_{jk} : bobot *node* j pada lapisan tersembunyi ke *node* k pada lapisan keluaran

V_{0k} : bias untuk *node* k pada lapisan keluaran

$f_1(\cdot)$: fungsi aktivasi pada lapisan tersembunyi

$\varepsilon_{k(t)}$: *error* pada *node* keluaran k dan pengamatan t

Gambar arsitektur FFNN meliputi variabel masukan yang diperoleh dari hasil pemodelan VAR (Gambar 2.3). Sedangkan penentuan banyaknya *node* pada lapisan tersembunyi diperoleh dari beberapa percobaan yang dilakukan oleh peneliti. Fungsi aktivasi yang digunakan pada lapisan tersembunyi adalah *logistic sigmoid*. Sedangkan pada lapisan keluaran menggunakan fungsi aktivasi linear. Hasil model FFNN terbaik ditentukan oleh nilai kriteria kebaikan model baik data *in-sample* maupun *out-of-sample*. Oleh karena itu, belum tentu semakin banyaknya *node* di lapisan masukan dan lapisan tersembunyi akan menjadi model FFNN terbaik.



Gambar 2.3 Arsitektur FFNN multi-output

2.6.1 Backpropagation

Backpropagation merupakan algoritma yang digunakan untuk melakukan *updating* bobot-bobot pada *Neural Network*. Pada algoritma *backpropagation* terdapat fase maju dan fase mundur. Pada fase maju, bobot-bobot dihitung mulai dari lapisan masukan sampai dengan keluaran dengan menggunakan fungsi aktivasi yang telah ditentukan. Kemudian diperoleh nilai error yang merupakan selisih antara nilai keluaran dengan nilai target. Sedangkan pada fase mundur, error tersebut dipropagasi mundur mulai dari lapisan keluaran sampai lapisan masukan, sehingga diperoleh bobot-bobot baru yang meminimumkan error (Siang, 2005).

Bobot dan bias yang menghubungkan lapisan masukan ke lapisan tersembunyi dinotasikan dengan w_{ij} dan w_{0j} , sedangkan bobot dan bias yang menghubungkan lapisan tersembunyi ke lapisan keluaran dinotasikan dengan v_{jk} dan v_{0k} . Langkah awal adalah menentukan nilai *learning rate* (η). Selanjutnya jika ditentukan nilai vektor variabel bebas \mathbf{x}_t sebagai nilai masukan dan vektor variabel terikat \mathbf{y}_t sebagai nilai target NN dimana diketahui $\mathbf{y}_t = (Y_{1(t)} \ Y_{2(t)} \ \dots \ Y_{K(t)})$ dan $\mathbf{x}_t = (X_{1(t)} \ \dots \ X_{K(t)} \ X_{K+1(t)} \ \dots \ X_{I(t)}) = (Y_{1(t)} \ \dots \ Y_{K(t-1)} \ Y_{1(t-2)} \ \dots \ Y_{K(t-d)})$. Sehingga

pada lapisan tersembunyi akan dihitung nilai estimasi keluaran sebagai berikut (Fithriasari, 2014).

$$\hat{Z}_{j(t)} = f_1 \left(\hat{W}_{0j} + \sum_{i=1}^I X_{i(t)} \hat{W}_{ij} \right), j = 1, \dots, J \quad (2.25)$$

dimana

\hat{W}_{ij} : estimasi bobot antara lapisan masukan dengan lapisan tersembunyi

\hat{W}_{0j} : estimasi bias untuk lapisan tersembunyi

$X_{i(t)}$: variabel masukan

$\hat{Z}_{j(t)}$: estimasi nilai keluaran pada lapisan tersembunyi

$f_1(\cdot)$: fungsi aktivasi pada lapisan tersembunyi

Selanjutnya dihitung nilai prediksi pada lapisan keluaran sebagai berikut

$$\hat{Y}_{k(t)} = f_2 \left(\hat{V}_{0k} + \sum_{j=1}^J \hat{Z}_j \hat{V}_{jk} \right), k = 1, \dots, K \quad (2.26)$$

dimana

$\hat{Y}_{k(t)}$: estimasi nilai prediksi pada lapisan keluaran

\hat{V}_{0k} : estimasi bias untuk lapisan keluaran

\hat{V}_{jk} : estimasi bobot antara lapisan tersembunyi dan lapisan keluaran

$f_2(\cdot)$: fungsi aktivasi pada lapisan keluaran

Langkah selanjutnya adalah menghitung koreksi bobot dan bias antara lapisan tersembunyi dan lapisan keluaran sebagai berikut

$$\Delta V_{jk} = \eta \hat{Z}_{j(t)} \delta_{k(t)}, \Delta V_{0k} = \eta \delta_{k(t)}, j = 1, \dots, J \text{ dan } k = 1, \dots, K \quad (2.27)$$

$$\delta_{k(t)} = (Y_{k(t)} - \hat{Y}_{k(t)}) f_2^* \left(\hat{V}_{0k} + \sum_{j=1}^J \hat{Z}_{j(t)} \hat{V}_{jk} \right), k = 1, \dots, K \quad (2.28)$$

dimana

η : *learning rate*

$\delta_{k(t)}$: *error* pada lapisan keluaran

Selanjutnya menghitung koreksi bobot dan bias antara lapisan masukan dan lapisan tersembunyi sebagai berikut.

$$\Delta W_{ij} = \eta X_{i(t)} \gamma_{j(t)}, \Delta W_{0j} = \eta \gamma_{j(t)}, i = 1, \dots, I \text{ dan } j = 1, \dots, J \quad (2.29)$$

dengan *error* pada lapisan tersembunyi adalah

$$\gamma_{j(t)} = \left(\sum_{k=1}^K \delta_{k(t)} \hat{V}_{jk} \right) f_1^* \left(\hat{W}_{0j} + \sum_{i=1}^I X_{i(t)} \hat{W}_{ij} \right), j = 1, \dots, J \quad (2.30)$$

Kemudian menghitung perbaruan nilai bobot dan bias sebagai berikut.

$$\hat{W}_{ij(\text{baru})} = \hat{W}_{ij(\text{lama})} + \Delta W_{ij} \text{ dan } \hat{V}_{jk(\text{baru})} = \hat{V}_{jk(\text{lama})} + \Delta V_{jk} \quad (2.31)$$

$$\hat{W}_{0j(\text{baru})} = \hat{W}_{0j(\text{lama})} + \Delta W_{0j} \text{ dan } \hat{V}_{0k(\text{baru})} = \hat{V}_{0k(\text{lama})} + \Delta V_{0k} \quad (2.32)$$

Selanjutnya menghitung nilai E sebagai berikut.

$$E = \frac{1}{2} \sum_{t=1}^n \sum_{k=1}^K \left\{ \left(Y_{k(t)} - \hat{Y}_{k(t)} \right)^2 \right\} \quad (2.33)$$

Langkah ini dilakukan secara berulang-ulang dan akan berhenti jika nilai E kurang dari batas yang telah ditentukan.

2.7 Metode Optimasi Levenberg Marquardt

Ada banyak metode yang dapat digunakan untuk optimasi, namun kelebihan dari metode Levenberg Marquardt ini stabil, *powerfull* dan lebih efisien karena lebih cepat mencapai konvergen (Kisi & Uncuoglu, 2005). Spesifikasi perbedaan beberapa metode akan ditampilkan dalam Tabel 2.2 sebagai berikut (Permai, 2013; Yu & Wilamowski, 2010)

Tabel 2.2 Spesifikasi Perbedaan Beberapa Metode Optimasi

Algoritma	Aturan Update	Konvergensi	Kompleksitas Komputasi
Gradient Descent	$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha g_k$	Stabil, lambat	Gradient
Newton	$\mathbf{w}_{k+1} = \mathbf{w}_k - \mathbf{H}_k^{-1} g_k$	Tidak stabil, cepat	Gradient dan Hessian
Algoritma	Aturan Update	Konvergensi	Kompleksitas Komputasi
Gauss-Newton	$\mathbf{w}_{k+1} = \mathbf{w}_k - \left(\mathbf{J}_k^T \mathbf{J}_k \right)^{-1} \mathbf{J}_k e_k$	Tidak stabil, cepat	Jacobian
Levenberg Marquadt	$\mathbf{w}_{k+1} = \mathbf{w}_k - \left(\mathbf{J}_k^T \mathbf{J}_k + \mu \mathbf{I} \right)^{-1} \mathbf{J}_k e_k$	Stabil, cepat	Jacobian

Levenberg Marquardt adalah salah satu metode optimasi yang digunakan untuk meminimumkan jumlah kuadrat error (*sum of square error*). Fungsi jumlah kuadrat error didefinisikan seperti berikut (Bishop, 1995).

$$E = \frac{1}{2} \sum_{t=1}^n \sum_{k=1}^K \hat{\varepsilon}_{k(t)}^2 \quad (2.34)$$

dimana $\hat{\varepsilon}_{k(t)}^2 = (Y_{k(t)} - \hat{Y}_{k(t)})^2$ dan k menunjukkan stasiun ke- k .

Apabila terdapat bobot lama (\mathbf{w}_{lama}) dan diperoleh bobot baru (\mathbf{w}_{baru}), serta errorsangat kecil, maka vektor error $\hat{\varepsilon}_{k(t)}$ dapat dijabarkan dengan deret Taylor pada order pertama dengan fungsi *error* seperti berikut.

$$\hat{\varepsilon}(\mathbf{w}_{\text{baru}}) = \hat{\varepsilon}(\mathbf{w}_{\text{lama}}) + \mathbf{J}(\mathbf{w}_{\text{baru}} - \mathbf{w}_{\text{lama}}) \quad (2.35)$$

$$E = \frac{1}{2} \left\| \hat{\varepsilon}(\mathbf{w}_{\text{lama}}) + \mathbf{J}(\mathbf{w}_{\text{baru}} - \mathbf{w}_{\text{lama}}) \right\|^2 \quad (2.36)$$

Nilai error ini diminimumkan dan dihubungkan dengan bobot baru (\mathbf{w}_{baru}) maka diperoleh bobot baru seperti berikut.

$$\mathbf{w}_{\text{baru}} = \mathbf{w}_{\text{lama}} - (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \hat{\varepsilon}(\mathbf{w}_{\text{lama}}) \quad (2.37)$$

dimana \mathbf{J} merupakan matrix Jacobian yaitu turunan pertama dari ε sedangkan matrix Hessian didefinisikan $\mathbf{H} = \mathbf{J}^T \mathbf{J}$. Pada prinsipnya *update* untuk bobot baru di atas dapat dilakukan dengan cara meminimumkan fungsi error, tetapi masalah yang muncul dengan pendekatan seperti di atas adalah nilainya dapat berubah relatif besar dalam kasus pendekatan linier. Pada algoritma Levenberg Marquardt dilakukan peminimuman fungsi error namun tetap membuat nilai errornya kecil sehingga pendekatan linier menjadi valid. Oleh karena itu, dilakukan modifikasi fungsi error sebagai berikut.

$$E = \frac{1}{2} \left\| \hat{\varepsilon}(\mathbf{w}_{\text{lama}}) + \mathbf{J}(\mathbf{w}_{\text{baru}} - \mathbf{w}_{\text{lama}}) \right\|^2 + \lambda \left\| \mathbf{w}_{\text{baru}} - \mathbf{w}_{\text{lama}} \right\|^2 \quad (2.38)$$

Apabila meminimumkan *error* yang telah dimodifikasi tersebut maka diperoleh fungsi bobot baru sebagai berikut.

$$\mathbf{w}_{\text{baru}} = \mathbf{w}_{\text{lama}} - (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I})^{-1} \mathbf{J}^T \varepsilon(\mathbf{w}_{\text{lama}}) \quad (2.39)$$

dimana λ adalah koefisien kombinasi yang nilainya selalu positif dan I adalah matriks identitas.

2.8 Bayesian Neural Network

Pada analisis bayesian semua parameter yang tidak diketahui dan diduga diperlakukan sebagai variabel random. Berbeda dengan pendekatan klasik, metode bayesian tidak melibatkan proses optimasi dalam inferensia, karena pendekatan bayesian mengaplikasikan teorema bayes yang didasarkan pada distribusi posterior gabungan dari seluruh parameter. Inferensia dengan pendekatan bayesian dilakukan dengan menggunakan distribusi posterior dari parameter. Oleh karena itu, tujuan utama metode bayesian adalah melakukan eksplorasi terhadap distribusi posterior tersebut. Metode bayesian akan melakukan estimasi parameter dengan menggunakan distribusi posterior marginal parameter tersebut. Distribusi posterior marginal ini diperoleh dengan cara mengintegalkan distribusi posterior gabungan. Pada tahap ini, khususnya untuk kasus model yang cukup kompleks umumnya timbul permasalahan yaitu proses integrasi tersebut menjadi sangat rumit dan sulit untuk memperoleh solusinya. Maka dalam hal ini cara yang digunakan dalam pendekatan bayesian adalah bukan dengan cara mengintegalkan distribusi posterior gabungan secara analitik, namun dengan menggunakan prosedur simulasi data yang mengikuti distribusi posterior gabungan dengan memanfaatkan bentuk *full conditional* untuk memperoleh distribusi posterior marginal setiap parameter yang akan diestimasi tersebut. Dengan demikian proses optimasi yang dilakukan dalam analisis klasik digantikan dengan proses integrasi pendekatan analisis bayesian (King, Morgan, Giemenez, & Brooks, 2010). Integrasi dalam analisis bayesian ini tidak dilakukan secara analitik terhadap distribusi posterior gabungan dari parameter, melainkan dengan pendekatan prosedur simulasi khusus yang menghasilkan sampel dari distribusi posterior tersebut. Proses ini yang dikenal dengan proses Markov Chain Monte Carlo (Koop, 2003).

Perspektif bayesian menyatakan bahwa data hasil pengamatan berasal dari suatu distribusi probabilitas yang memiliki parameter-parameter yang tidak diketahui dengan pasti. Oleh karena itu, perlu ditentukan suatu distribusi dari parameter tersebut yang disebut dengan distribusi prior. Kombinasi antara prior dari

parameter dengan informasi data sampel akan menghasilkan distribusi posterior dari parameter. Distribusi posterior ini menyatakan pola ketidakpastian dari nilai parameter populasi setelah diperoleh data hasil pengamatan. Distribusi posterior merupakan paduan antara data observasi yang digunakan untuk menyusun fungsi likelihood dengan informasi masa lalu sebelum dilakukan observasi, sehingga dapat ditulis seperti berikut (Box & Tiao, 1992).

distribusi posterior \propto likelihood x distribusi prior

Model FFNN mempunyai parameter $\{v_{0k}, w_{0j}, \mathbf{v}_{jk}, \mathbf{w}_{ij}, \tau\}$, dimana v_{0k} adalah bias node keluaran, w_{0j} adalah bias node tersembunyi ke- j , \mathbf{v}_{jk} adalah vektor bobot antara node tersembunyi ke- j dan lapisan keluaran ke- k , \mathbf{w}_{ij} adalah vektor bobot antara lapisan masukan ke- i dan lapisan tersembunyi ke- j . Prior untuk v_{0k} dan w_{0j} adalah berdistribusi normal, sedangkan parameter \mathbf{v}_{jk} dan \mathbf{w}_{ij} berdistribusi normal multivariat (Fithriasari, 2014; Permai, 2013; Muller&Insua, 1998), parameter τ berdistribusi gamma, sehingga dapat ditulis sebagai berikut.

$$\begin{aligned}
v_{0k} &\sim N(\mu_{v_{0k}}, \sigma_{v_{0k}}^2), \quad k = 1, 2 \\
\mathbf{v}_{jk} &\sim N_J(\boldsymbol{\mu}_{\mathbf{v}_{jk}}, \boldsymbol{\Sigma}_{\mathbf{v}_{jk}}), \quad j = 1, 2 \quad k = 1, 2 \\
w_{0j} &\sim N(\mu_{w_{0j}}, \sigma_{w_{0j}}^2), \quad j = 1, 2 \\
\mathbf{w}_{ij} &\sim N_I(\boldsymbol{\mu}_{\mathbf{w}_{ij}}, \boldsymbol{\Sigma}_{\mathbf{w}_{ij}}), \quad i = 1, 2 \quad k = 1, 2 \\
\tau^{-1} &\sim \text{Gamma}(a, b)
\end{aligned} \tag{2.40}$$

Ketika ada ketidakpastian yang tidak dapat diabaikan tentang *prior hyperparameter*, maka *prior* model dapat dikembangkan dengan penambahan *hyperpriors* sebagai berikut.

$$\begin{aligned}
\boldsymbol{\mu}_{\mathbf{v}_{jk}} &\sim N(\mathbf{a}_{\mathbf{v}_{jk}}, \mathbf{B}_{\mathbf{v}_{jk}}^{-1}) \\
\boldsymbol{\mu}_{\mathbf{w}_{ij}} &\sim N(\mathbf{a}_{\mathbf{w}_{ij}}, \mathbf{B}_{\mathbf{w}_{ij}}^{-1}) \\
\boldsymbol{\Sigma}_{\mathbf{v}_{jk}}^{-1} &\sim \text{Wishart}(\mathbf{A}_{\mathbf{v}_{jk}}, b_{\mathbf{v}_{jk}}) \\
\boldsymbol{\Sigma}_{\mathbf{w}_{ij}}^{-1} &\sim \text{Wishart}(\mathbf{A}_{\mathbf{w}_{ij}}, b_{\mathbf{w}_{ij}})
\end{aligned} \tag{2.41}$$

Jika $\Theta = (\mu_v, \Sigma_v, \mu_w, \Sigma_w)$ adalah himpunan *hyperparameter* dan $\theta = (w, v, \tau)$ adalah himpunan parameter model FFNN, maka distribusi prior gabungan parameter adalah

$$p(w, v, \tau, \Theta) = p(\mu_v, \Sigma_v, \mu_w, \Sigma_w) p(\tau) \prod_{j=1}^2 \prod_{k=1}^2 p(v_{jk} | \mu_v, \Sigma_v) \prod_{i=1}^2 \prod_{j=1}^2 p(w_{ij} | \mu_w, \Sigma_w) \quad (2.42)$$

Apabila diberikan sekelompok data $D = \{y_1, y_2, \dots, y_n\}$ dan θ merupakan vektor parameter model, maka fungsi *likelihood* dari data D adalah (Fithriasari, 2014; Menchero, Diez, & Insua, 2005).

$$\begin{aligned} p(D|\theta) &= p(y_1, y_2, \dots, y_d | \theta) p(y_{d+1}, y_{d+2}, \dots, y_n | y_1, y_2, \dots, y_d, \theta) \\ &= p(y_1, y_2, \dots, y_d | \theta) p(D^* | y_1, y_2, \dots, y_d, \theta) \end{aligned}$$

dimana $D^* = (y_{d+1}, y_{d+2}, \dots, y_n)$ dan

$$\begin{aligned} p(D^* | y_1, y_2, \dots, y_d, \theta) &= \prod_{t=d+1}^n p(y_t | y_{t-1}, y_{t-2}, \dots, y_{t-d}, \theta) \\ &= \prod_{t=d+1}^n p(y_t | y_{t-1}, y_{t-2}, \dots, y_{t-d}, w, v, \tau) \end{aligned} \quad (2.43)$$

Selanjutnya untuk mendapatkan distribusi peluang *posterior* parameter $p(\theta|D)$, maka digunakan teorema *Bayes* sebagai berikut (Fithriasari, 2014).

$$P(\theta|D) = \frac{P(D|\theta)}{P(D)} \quad (2.44)$$

dimana $P(\theta|D)$ adalah fungsi *likelihood* sedangkan $P(D)$ merupakan *normalization factor* yang dapat dihitung sebagai berikut.

$$P(D) = \int_{-\infty}^{\infty} P(D|\theta) P(\theta) d\theta$$

Karena *normalization factor* tidak bergantung pada θ , maka $P(D)$ dapat dianggap sebagai konstanta sehingga dapat diabaikan maka distribusi *posterior* dapat ditulis sebagai berikut.

$$P(\theta|D) \propto P(D|\theta) P(\theta) \quad (2.45)$$

Jika distribusi *prior* seperti pada persamaan (2.41) dan (2.42), maka distribusi *posterior* gabungan adalah sebagai berikut.

$$P(\boldsymbol{\theta}, \Theta | D) \propto P(D^* | y_1, y_2, \dots, y_d, \boldsymbol{\theta}) p(w, v, \tau, \Theta) \quad (2.46)$$

Hasil distribusi keluaran *neural network* akan berupa distribusi prediktif posterior keluaran y_t^* dengan syarat telah diberikan nilai masukan baru x_t^* dan data $D^* = (y_{d+1}, y_{d+2}, \dots, y_n)$. Distribusi prediktif keluaran ini dapat diperoleh dengan mengintegrasikan model prediksi distribusi posteriornya (Fithriasari, 2014; Lampinen & Vehtari, 2001).

$$p(y_t^* | \mathbf{x}_t^*, D^*) = \int p(y_t^* | \mathbf{x}_t^*, \Phi) p(\Phi | D^*) d\Phi \quad (2.47)$$

dimana $\Phi = (\boldsymbol{\theta}, \Theta)$ adalah himpunan parameter model dan *hyperparameter*, dan $P(\Phi | D^*)$ adalah distribusi posterior dari persamaan (2.46).

Model dalam persamaan (2.47) terlihat sangat kompleks, sehingga untuk menyelesaikan persamaan tersebut dibutuhkan analisis yang cukup rumit, karena melibatkan integrasi multidimensi dan persamaan nonlinier. Hal tersebut dapat diatasi dengan pendekatan numerik yaitu metode *Markov Chain Monte Carlo* (MCMC).

2.9 Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo (MCMC) merupakan metode integrasi untuk mendapatkan model yang sesuai dimana memiliki kompleksitas tak terbatas (Charlin & Chib, 1995). Untuk mendapatkan estimasi parameter Bayesian *Neural Network* dari distribusi posterior melalui proses integrasi seringkali sulit dilakukan apabila melibatkan persamaan integral yang sangat kompleks. Oleh karena itu, penyelesaian perhitungan estimasi parameter seringkali dilakukan secara numerik, salah satunya adalah metode Markov Chain Monte Carlo (MCMC). Pendekatan integrasi Monte Carlo sangat efektif untuk mengurangi beban komputasi dalam menyelesaikan integrasi secara numerik (Carlin, 1992). MCMC sangat berguna dalam mengestimasi suatu parameter yang terkadang sangat rumit dalam proses integrasi dan sulit diselesaikan secara analitis (Iriawan, Astutik, & Prasetyo, 2010).

Metode ini memungkinkan proses simulasi dengan mengambil sampel acak dari model stokastik yang sangat rumit (Scollnik, 2011).

Algoritma yang sering digunakan sebagai pembangkit variabel random dalam MCMC adalah *Gibbs Sampling* (Casella & George, 1992). *Gibbs Sampling* merupakan suatu teknik simulasi untuk membangkitkan suatu variabel random dari suatu distribusi tertentu secara langsung. Penggunaan *Gibbs Sampling* pada suatu data ditujukan untuk mendapatkan data bangkitan tiap parameter. *Gibbs Sampler* yang digunakan disini akan menirukan proses markov secara iteratif dengan melakukan estimasi parameter model yang hanya dengan menggunakan informasi nilai parameter hasil estimasi dari suatu langkah iterasi sebelumnya. Sehingga proses *Gibbs Sampling* akan membangun langkah-langkah proses MCMC dengan membuat *full conditional posterior* untuk setiap posterior parameter kemudian menyusunnya secara bergantian sebagai langkah iteratif simulasi stokastik. Pada beberapa kasus, terlihat bahwa cara kerja *Gibbs Sampling* hanya didasarkan pada sifat dasar rantai markov. MCMC dilakukan dengan cara membangkitkan data pada parameter θ menggunakan Gibbs Sampler. Parameter θ dianggap sebagai vektor random dengan distribusi tertentu (Box & Tiao, 1992). Algoritma dalam MCMC untuk mendapatkan posterior, yaitu sebagai berikut (Fithriasari, 2014).

1. Menentukan inisial nilai awal Θ dan τ
2. Untuk setiap w_{ij} , $i=1,...,I$ $j=0,...,J$, bangkitkan $\tilde{w}_{ij} \sim N(\mu_{w_{ij}}, \Sigma_{w_{ij}})$
3. Bangkitkan nilai baru setiap parameter berdasarkan *full conditional posterior*

$$\tilde{\mathbf{v}} \sim P(\mathbf{v} | \mathbf{w}, \tau, \Theta, D^*)$$

$$\tilde{\tau} \sim P(\tau | \mathbf{w}, \mathbf{v}, \Theta, D^*)$$
4. Berdasarkan nilai yang telah dibangkitkan dari (w, v, σ^2) , kemudian bangkitkan nilai baru untuk setiap *hyperparameter* dengan penarikan sampel berdasarkan *full conditional posterior distribution* sebagai berikut.

$$\tilde{\mu}_{\mathbf{v}} \sim P(\mu_{\mathbf{v}} | \mathbf{v}, \Sigma_{\mathbf{v}}, D^*)$$

$$\tilde{\Sigma}_{\mathbf{v}} \sim P(\Sigma_{\mathbf{v}} | \mathbf{v}, \mu_{\mathbf{v}}, D^*)$$

$$\tilde{\mu}_{\mathbf{w}} \sim P(\mu_{\mathbf{w}} | \mathbf{w}, \Sigma_{\mathbf{w}}, D^*)$$

$$\tilde{\Sigma}_{\mathbf{w}} \sim P(\Sigma_{\mathbf{w}} | \mathbf{w}, \mu_{\mathbf{w}}, D^*)$$

5. Ulangi langkah (2) sampai dengan (4) sebanyak m kali untuk mendapatkan sebanyak m sampel yang diinginkan.

Sifat Markov Chain yang *strongly ergodic* dapat diindikasikan melalui diagnostik plot yang diperoleh dari proses MCMC, yaitu (Ntzoufras, 2009):

- a. *Trace Plot*, yaitu plot dari iterasi dan nilai sampel yang diperoleh untuk setiap iterasi. Apabila semua nilai sampel yang diperoleh menunjukkan plot yang acak dan secara stabil berada dalam suatu domain/zona nilai dan tidak terdapat pola periodik yang kuat maka dapat dinyatakan bahwa telah memenuhi sifat MCMC yang *strongly ergodic* atau kondisi *equilibrium* telah tercapai. *Trace plot* ini menggambarkan pergerakan nilai sampel yang diperoleh untuk sebagian iterasi saja, sedangkan pergerakan nilai sampel untuk seluruh iterasi digambarkan dengan serial plot.
- b. *Autocorrelation plot*, yaitu plot dari nilai *Autocorrelation Function* (ACF). Apabila nilai ACF yang diperoleh sangat kecil sehingga mendekati nol maka dapat dinyatakan bahwa sampel parameter yang diperoleh telah memenuhi sifat MCMC yang *strongly ergodic* atau kondisi *equilibrium* telah tercapai.

Kondisi *equilibrium* dinyatakan tercapai apabila nilai *ergodic mean* yang diperoleh dalam proses estimasi parameter telah menunjukkan nilai yang stabil dan berada dalam 95% *credible interval*. *Ergodic mean* adalah rata-rata bergerak dari estimasi posterior parameter selama proses iterasi berlangsung.

2.10 Kriteria Model Terbaik

Kriteria pemilihan model terbaik berdasarkan pada kriteria *in-sample* dan *out-of-sample*. Kriteria yang digunakan untuk *in-sample* adalah *Akaike's Information Criterion* (AIC). Perhitungan untuk AIC diperoleh dari rumus berikut (Permai, 2013; Wei, 2006).

$$AIC = n \ln \left| \hat{\Sigma}_p \right| + 2pm^2 \quad (2.48)$$

dengan:

$$\hat{\Sigma}_p = \frac{1}{n} (\hat{\mathbf{\epsilon}} \hat{\mathbf{\epsilon}}') \quad ; \quad \hat{\mathbf{\epsilon}} = \begin{pmatrix} \hat{\epsilon}_1 \\ \hat{\epsilon}_2 \\ \vdots \\ \hat{\epsilon}_n \end{pmatrix}$$

dimana p adalah banyaknya parameter yang diestimasi, n adalah banyaknya data *in-sample* untuk tiap variabel respon, m adalah banyaknya variabel respon (dalam penelitian ini adalah banyaknya node di lapisan keluaran).

Sedangkan untuk kriteria *out-of-sample* terdapat dua kriteria yaitu *Mean Absolute Percentage Error* (MAPE) dan *Root Mean Square Error* (RMSE) untuk tiap variabel respon dengan rumus sebagai berikut.

$$\text{MAPE}_k = \frac{\sum_{t=1}^n \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right|}{n} \times 100\% \quad (2.49)$$

$$\text{RMSE}_k = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2} \quad (2.50)$$

dengan n adalah banyaknya data *out-of-sample*, \hat{Y}_t adalah hasil nilai ramalan, Y_t adalah nilai sebenarnya, dan k menyatakan lokasi/stasiun. Nilai MAPE dan RMSE yang semakin kecil maka semakin baik model tersebut karena nilai error dari hasil ramalan semakin kecil.

2.11 Geografi Semarang

Semarang adalah ibu kota provinsi Jawa Tengah sekaligus kota metropolitan terbesar kelima di Indonesia setelah Jakarta, Surabaya, Bandung, dan Medan (Wikipedia, 2015). Kota Semarang mempunyai posisi astronomi diantara garis 6° 50' - 7° 10' Lintang Selatan dan garis 109° 35' - 110° 50' Bujur Timur. Luas wilayah kota Semarang mencapai 373,70 km² dengan 16 kecamatan dan 177 kelurahan. Secara topografi terdiri atas daerah pantai, dataran rendah dan perbukitan, sehingga memiliki wilayah yang disebut sebagai kota bawah dan kota atas. Curah hujan di Kota Semarang mempunyai sebaran yang tidak merata sepanjang tahun, dengan total curah hujan rata-rata 9.891 mm per tahun.

Suhu minimum rata-rata yang diukur di Stasiun Klimatologi Semarang berubah-ubah dari 21,1°C pada September ke 24,6°C pada bulan Mei, dan suhu maksimum rata-rata berubah-ubah dari 29,9°C ke 32,9°C. Kelembaban relatif bulanan rata-rata berubah-ubah dari minimum 61% pada bulan September ke maksimum 83% pada bulan Januari. Kecepatan angin bulanan rata-rata di Stasiun Klimatologi Semarang berubah-ubah dari 215 km/hari pada bulan Agustus sampai 286 km/hari pada bulan Januari. Lamanya sinar matahari yang menunjukkan rasio sebenarnya sampai lamanya sinar matahari maksimum hari, bervariasi dari 46% pada bulan Desember sampai 98% pada bulan Agustus.

2.12 Temperatur

Temperatur adalah keadaan udara pada waktu dan tempat tertentu. Beberapa hal yang mempengaruhi temperatur adalah sebagai berikut.

- a. Penyinaran matahari. Jarak dan lamanya penyinaran serta besar kecilnya sudut datang matahari. Semakin besar intensitas penyinaran matahari maka semakin tinggi temperaturnya.
- b. Tinggi rendahnya permukaan daratan. Semakin tinggi suatu tempat di daratan dari permukaan air laut maka semakin rendah temperaturnya.
- c. Letak lintang.
- d. Sifat permukaan bumi dimana permukaan daratan lebih cepat menyerap dan melepas panas dibandingkan permukaan lautan.

Alat untuk mengukur temperatur adalah termometer (Samadi, 2006). Suhu udara untuk kenyamanan tubuh manusia yaitu antara 22°C sampai dengan 27°C. Semakin tinggi beban kerja yang diberikan kepada pekerja, maka semakin besar pengaruh kelembaban dan temperatur terhadap perubahan laju detak jantung pekerja, sehingga pada suhu 22°C dengan berapapun tingkat kelembaban relatifnya pekerja tetap dapat bekerja secara optimal (Purnomo & Rizal, 2000).

(Halaman ini sengaja dikosongkan)

BAB III

METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian ini adalah data sekunder dari Badan Meteorologi, Klimatologi dan Geofisika (BMKG). Pengukuran data iklim dan cuaca di kota Semarang dilakukan pada stasiun Semarang dan stasiun Ahmad Yani. Variabel yang digunakan dalam penelitian ini adalah temperatur rata-rata tiap hari pada kedua stasiun di kota Semarang tahun 2013 dengan satuan derajat celcius ($^{\circ}\text{C}$). Data yang akan dianalisis dibagi menjadi data *in-sample* dan *out-of-sample*. Data *in-sample* yang digunakan sebanyak 243 data dan *out-of-sample* sebanyak 30 data. Struktur data pada penelitian ini adalah sebagai berikut.

Tabel 3.1 Struktur Data Penelitian			
Bulan	Hari ke-	Y_1	Y_2
Januari	1	$Y_{1,1}$	$Y_{2,1}$
	2	$Y_{1,2}$	$Y_{2,2}$
	\vdots		
	31	$Y_{1,31}$	$Y_{2,31}$
Februari	32	$Y_{1,32}$	$Y_{2,32}$
	33	$Y_{1,33}$	$Y_{2,33}$
	\vdots		
	59	$Y_{1,59}$	$Y_{2,59}$
\vdots	\vdots	\vdots	\vdots
Agustus	213	$Y_{1,213}$	$Y_{2,213}$
	214	$Y_{1,214}$	$Y_{2,214}$
	\vdots		
	243	$Y_{1,243}$	$Y_{2,243}$

3.2 Metode Analisis

Metode dan langkah-langkah analisis untuk mencapai tujuan penelitian ini adalah

1. Eksplorasi data
 - a. Deskripsi pola temperatur kota Semarang di stasiun Semarang dan Ahmad Yani.
 - b. Melakukan identifikasi secara *univariate* dengan membentuk *timeseries plot* dan Box-Cox.
 - c. Melakukan identifikasi secara *multivariate* dengan membentuk plot MACF dan MPACF.
2. Pemodelan temperatur di kota Semarang menggunakan *Vector Autoregressive* (VAR) dengan langkah-langkah sebagai berikut.
 - a. Mendapatkan model sementara dengan model VAR.
 - b. Melakukan penaksiran parameter model VAR.
 - c. Menguji signifikansi parameter-parameter pada model VAR.
 - d. Melakukan pengujian asumsi pada residual yaitu asumsi residual *white noise* dan berdistribusi normal *multivariate*.
3. Melakukan uji deteksi hubungan *nonlinearitas* pada kedua stasiun dengan menggunakan uji *terasvirta*.
4. Penaksiran parameter pemodelan temperatur di kota Semarang menggunakan arsitektur *Feed Forward Neural Network* dengan algoritma *Backpropagation*, dengan langkah-langkah sebagai berikut.
 - a. Menentukan arsitektur *Feed Forward Neural Network* (FFNN) dari pemodelan data temperatur dengan *multi-output*.
 - b. Menentukan *input* berdasarkan pemodelan VAR.
 - c. Menentukan *learning rate*.
 - d. Inisialisasi parameter (bobot dan bias) pada model NN.
 - e. Menghitung estimasi nilai keluaran pada lapisan tersembunyi.
 - f. Menghitung nilai prediksi pada lapisan keluaran.
 - g. Menghitung *error* pada lapisan keluaran.
 - h. Menghitung koreksi bobot dan bias antara lapisan tersembunyi dan lapisan keluaran.

- i. Menghitung *error* pada lapisan tersembunyi.
 - j. Menghitung koreksi bobot dan bias antara lapisan masukan dan lapisan tersembunyi.
 - k. *Update* parameter (bobot dan bias) menggunakan algoritma *backpropagation* dengan metode optimasi *levenberg marquadt*.
 - l. Menghitung nilai *E* sampai nilai tersebut kurang dari batas yang diinginkan.
5. Penaksiran parameter pemodelan temperatur di kota Semarang menggunakan pendekatan Bayesian *Neural Network* dengan langkah-langkah sebagai berikut.
- a. Pemeriksaan distribusi data temperatur di kota Semarang pada masing-masing stasiun Semarang dan Ahmad Yani.
 - b. Menentukan *input* berdasarkan hasil pemodelan FFNN.
 - c. Melakukan estimasi parameter (bobot dan bias) menggunakan pendekatan Bayesian dengan cara berikut.
 - i. Estimasi parameter antara lapisan masukan dengan lapisan tersembunyi.
 - 1. Memilih distribusi prior yang sesuai dengan parameter *network*.
 - 2. Menentukan distribusi prior gabungan parameter.
 - 3. Menentukan fungsi likelihood.
 - 4. Menentukan distribusi posterior gabungan.
 - 5. Estimasi parameter-parameter *network* menggunakan MCMC dengan Gibbs Sampler.
 - ii. Membuat fungsi *network* pada tiap node di lapisan tersembunyi.
 - iii. Menghitung nilai dari fungsi aktivasi pada tiap node di lapisan tersembunyi.
 - iv. Estimasi parameter antara lapisan tersembunyi dengan lapisan keluaran.
 - 1. Memilih distribusi prior yang sesuai dengan parameter *network*.
 - 2. Menentukan distribusi prior gabungan parameter.
 - 3. Menentukan fungsi likelihood.
 - 4. Menentukan distribusi posterior gabungan.
 - 5. Estimasi parameter-parameter *network* menggunakan MCMC dengan Gibbs Sampler.
 - v. Membuat fungsi *network* pada tiap *node* di lapisan keluaran.

- vi. Menghitung nilai dari fungsi aktivasi berdasarkan *network* pada tiap node di lapisan keluaran.
 - d. Membuat model temperatur di kota Semarang dari hasil estimasi parameter menggunakan *Neural Network* pendekatan Bayesian.
 - e. Menghitung hasil ramalan.
6. Membandingkan hasil kinerja pemodelan VAR, FFNN dan BNN untuk kasus data temperatur di Semarang dengan kriteria model terbaik.

BAB IV

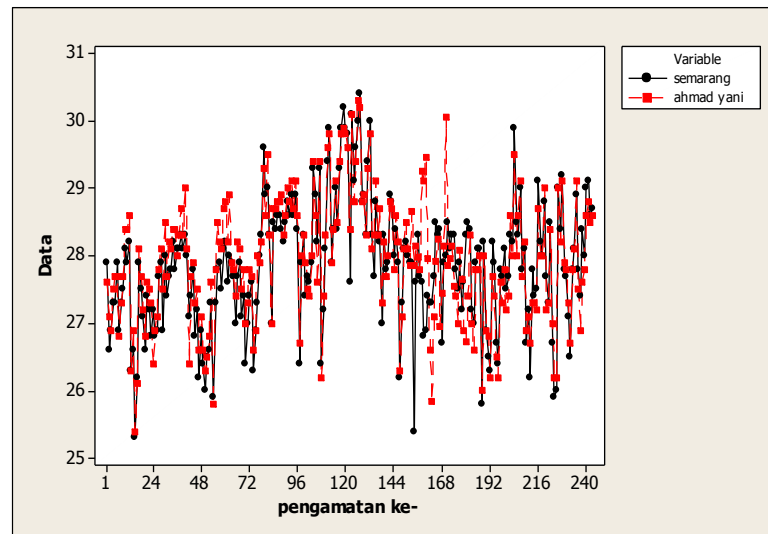
ANALISIS DAN PEMBAHASAN

4.1 Statistika Deskriptif

Karakteristik dari temperatur harian kota Semarang pada bulan Januari hingga Agustus tahun 2013 dapat dilihat dari analisis statistika deskriptif. Hal ini dilakukan untuk mengetahui gambaran umum kondisi temperatur Semarang selama 8 bulan terakhir. Analisis statistika deskriptif dapat dilihat pada Tabel 4.1 sebagai berikut.

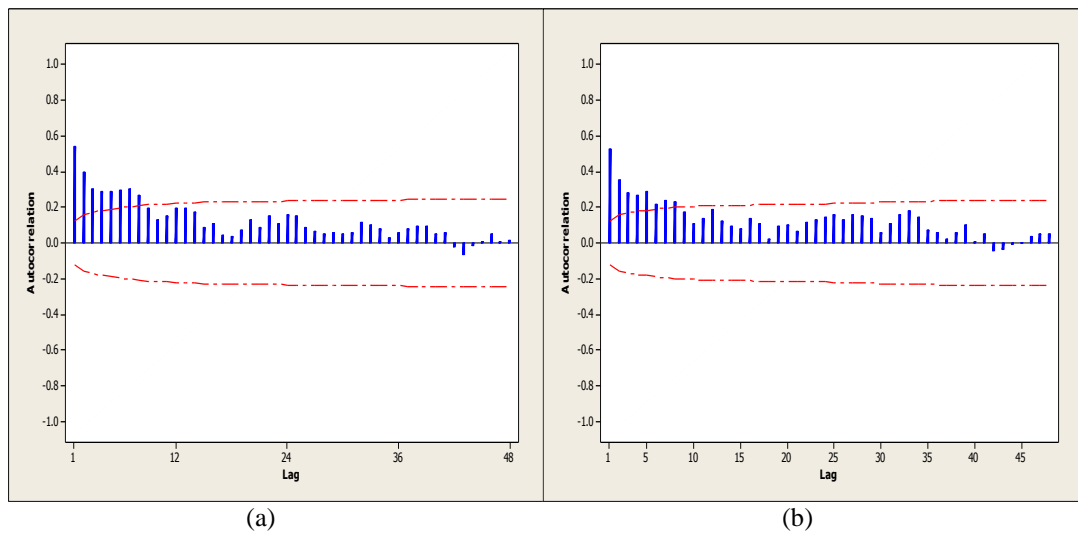
Tabel 4.1 Statistika Deskriptif

Variabel	Rata-rata	Median	Varians	Minimum	Maksimum
Semarang	27,852	27,90	0,861	25,3	30,4
Ahmad Yani	27,934	27,95	0,859	25,4	30,3

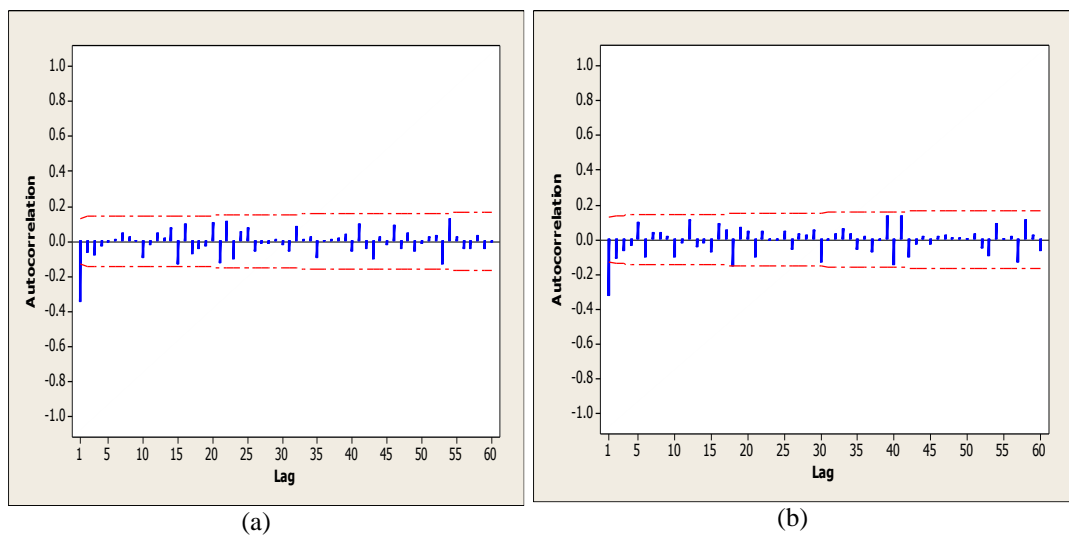


Gambar 4.1 Time series plot temperatur Semarang di stasiun Semarang dan Ahmad Yani

Fluktuasi data untuk stasiun Semarang sedikit lebih besar dibandingkan dengan stasiun Ahmad Yani (Gambar 4.1). Hal ini sesuai pada Tabel 4.1 dimana nilai varians untuk stasiun Semarang sedikit lebih besar daripada stasiun Ahmad Yani. Hasil dari plot ACF dapat diketahui bahwa lag menurun secara lambat sehingga dapat disimpulkan data belum stasioner dalam rata-rata maka perlu dilakukan *differencing* (Gambar 4.2). Setelah dilakukan pengecekan secara visual, agar lebih yakin maka dilakukan uji *t* yang dijelaskan pada Tabel 4.2.



Gambar 4.2 *Autocorrelation Function (ACF)* di stasiun (a) Semarang dan (b) Ahmad Yani sebelum *differencing*



Gambar 4.3 *Autocorrelation Function (ACF)* di stasiun (a) Semarang dan (b) Ahmad Yani setelah *differencing*

Tabel 4.2 Hasil Uji *t* masing-masing stasiun sebelum *differencing*

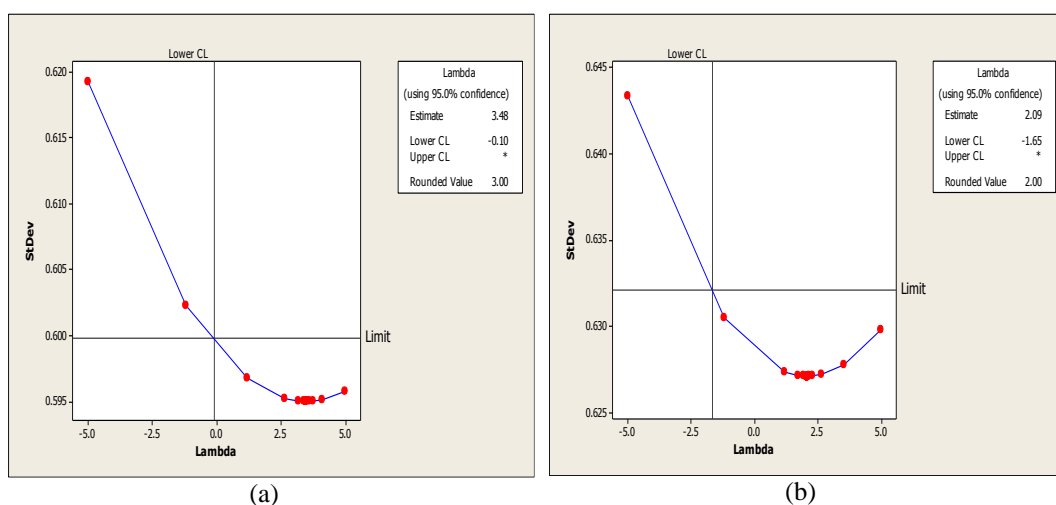
Variabel	<i>t</i>	P_value
Semarang	468,04	0,000
Ahmad Yani	469,81	0,000

Tabel 4.3 Hasil Uji *t* masing-masing stasiun setelah *differencing*

Variabel	<i>t</i>	P_value
Semarang	0,06	0,954
Ahmad Yani	0,07	0,943

Berdasarkan Tabel 4.2 dapat diperoleh keputusan bahwa tolak H_0 karena nilai $t > t_{0,025;242}$ atau $p_value < \alpha$ dengan α sebesar 5% dimana nilai $t_{0,025;242} = 1,96982$. Sehingga dapat dikatakan bahwa data belum stasioner terhadap *mean*. Maka perlu

dilakukan *differencing* sehingga memperoleh keputusan bahwa gagal tolak H_0 karena nilai $t < t_{0,025;242}$ atau $p_value > \alpha$ dengan α sebesar 5% dimana nilai $t_{0,025;242} = 1,96982$ (Tabel 4.3). Sehingga hasil uji t dengan *differencing* menunjukkan bahwa data telah stasioner terhadap rata-rata. Apabila dilihat dari plot ACF juga dapat diketahui bahwa lag menurun secara cepat (Gambar 4.3). Selanjutnya dilakukan pendeteksian stasioner dalam varians menggunakan Box-Cox pada Gambar 4.4.



Gambar 4.4 Box-Cox plot temperatur di stasiun (a) Semarang dan (b) Ahmad Yani

Data temperatur di stasiun Semarang dan Ahmad Yani telah menunjukkan bahwa sudah stasioner dalam varians secara *univariate* (Gambar 4.4). Karena nilai batas atas tidak diketahui untuk kedua stasiun sehingga data tidak perlu untuk ditransformasi.

4.2 Vector Autoregressive (VAR)

Langkah pertama yang dilakukan pada pemodelan VAR adalah menentukan orde dari data yang sudah stasioner. Pemeriksaan stasioneritas data temperatur di stasiun Semarang dan Ahmad Yani untuk kasus multivariat dapat menggunakan plot MPACF. Apabila hasil dari plot MPACF banyak mempunyai tanda positif maka perlu dilakukan *differencing*. Hasil plot MPACF pada data temperatur kota Semarang di stasiun Semarang dan Ahmad Yani sebelum dan setelah *differencing* dapat ditampilkan pada Gambar 4.6 dan 4.7 sebagai berikut.

Name/Lag	0	1	2	3	4	5	6	7	8	9	10
y1	++	++	++	++	++	++	++	++	++	++	..
y2	++	++	++	++	++	++	++	++	++	++	..
+ is > 2*std error, - is < -2*std error, . is between											

Gambar 4.5 Plot MACF di stasiun Semarang dan Ahmad Yani sebelum *differencing*

Name/Lag	1	2	3	4	5	6	7	8	9	10
y1	+.	+.
y2	+.
+ is > 2*std error, - is < -2*std error, . is between										

Gambar 4.6 Plot MPACF di stasiun Semarang dan Ahmad Yani sebelum *differencing*

Berdasarkan Gambar 4.5 dan 4.6 menunjukkan bahwa data pada kedua stasiun belum stasioner. Hal ini ditunjukkan dari masih banyak tanda positif pada lag 1 sampai 9 untuk plot MACF sedangkan lag 1 dan 7 untuk plot MPACF. Oleh karena itu, dilakukan *differencing* lag pertama untuk mengatasi data yang belum stasioner dengan hasil sebagai berikut.

Name/Lag	1	2	3	4	5	6	7	8	9	10
y1	-. .	:-	:-	:-
y2	-. .	:-	:-	:-
+ is > 2*std error, - is < -2*std error, . is between										

Gambar 4.7 Plot MPACF di stasiun Semarang dan Ahmad Yani setelah *differencing* 1

Hasil identifikasi plot MPACF setelah *differencing* 1 pada Gambar 4.7 menunjukkan bahwa lag 1, 2, 3 dan lag 6 signifikan. Karena terdapat tanda negatif pada lag 1, 2, 3 dan lag 6. Selanjutnya dilakukan identifikasi model VAR berdasarkan lag-lag yang signifikan. Tidak semua parameter memiliki pengaruh signifikan pada model sehingga perlu dilakukan proses *restrict*. Proses *restrict* dilakukan dengan cara mengeluarkan satu per satu variabel yang tidak signifikan dari model secara bertahap. Variabel pertama yang dikeluarkan adalah variabel dengan nilai *p_value* paling tinggi, sampai dengan diperoleh semua variabel signifikan pada model dengan nilai α sebesar 0,05. Sehingga hasil yang diperoleh dari Tabel 4.5 merupakan hasil *restrict*. Setelah itu, diperoleh nilai AIC dari hasil *restrict* pada masing-masing model yang dapat ditunjukkan pada Tabel 4.4 sebagai berikut.

Tabel 4.4 Nilai AIC model VAR

Model	AIC
VAR (1)	-1,78875
VAR (2)	-1,81327
VAR (3)	-1,95679
VAR (6)	-1,95237

Nilai AIC terkecil terdapat pada model VAR (3) sebesar -1,95679 (Tabel 4.4). Terdapat 7 parameter yang signifikan pada model VARIMA (3,1,0) dengan estimasi parameter disajikan pada Tabel 4.5. Jadi, dapat disimpulkan bahwa semua parameter yang diestimasi memiliki pengaruh yang signifikan terhadap model.

Tabel 4.5 Estimasi parameter model VARIMA (3,1,0)

Stasiun	Parameter	Estimasi	Std Error	t	p_value	Variabel
Semarang (Y_{1t}^*)	ϕ_{111}	-0,40427	0,04816	-8,39	0,0001	$Y_{1(t-1)}^*$
	ϕ_{211}	-0,27774	0,05092	-5,46	0,0001	$Y_{1(t-2)}^*$
	ϕ_{311}	-0,19610	0,06143	-3,19	0,0016	$Y_{1(t-3)}^*$
Ahmad Yani (Y_{2t}^*)	ϕ_{122}	-0,419	0,04789	-8,75	0,0001	$Y_{2(t-1)}^*$
	ϕ_{222}	-0,26876	0,05015	-5,36	0,0001	$Y_{2(t-2)}^*$
	ϕ_{321}	0,18332	0,07715	2,38	0,0183	$Y_{1(t-3)}^*$
	ϕ_{322}	-0,35457	0,05977	-5,93	0,0001	$Y_{2(t-3)}^*$

Berdasarkan hasil estimasi parameter pada Tabel 4.5 maka bentuk umum model VARIMA (3,1,0) adalah sebagai berikut.

$$\begin{bmatrix} Y_{1,t}^* \\ Y_{2,t}^* \end{bmatrix} = \begin{bmatrix} -0,40427 & 0 \\ 0 & -0,419 \end{bmatrix} \begin{bmatrix} Y_{1,t-1}^* \\ Y_{2,t-1}^* \end{bmatrix} + \begin{bmatrix} -0,27774 & 0 \\ 0 & -0,26876 \end{bmatrix} \begin{bmatrix} Y_{1,t-2}^* \\ Y_{2,t-2}^* \end{bmatrix} + \begin{bmatrix} -0,19610 & 0 \\ 0,18332 & -0,35457 \end{bmatrix} \begin{bmatrix} Y_{1,t-3}^* \\ Y_{2,t-3}^* \end{bmatrix} + \begin{bmatrix} a_{1t} \\ a_{2t} \end{bmatrix}$$

dikarenakan variabel $Y_{1,t}^*$ merupakan hasil *differencing* 1 maka

$$Y_{i,t}^* = (1-B)Y_{i,t} = Y_{i,t} - Y_{i,t-1}$$

Persamaan model VARIMA (3,1,0) pada masing-masing lokasi stasiun yaitu Semarang dan Ahmad Yani akan dijelaskan sebagai berikut.

$$\begin{aligned}
Y_{1t} &= Y_{1(t-1)} - 0,40427Y_{1(t-1)} + 0,40427Y_{1(t-2)} - 0,27774Y_{1(t-2)} + 0,27774Y_{1(t-3)} - \\
&\quad 0,19610Y_{1(t-3)} + 0,19610Y_{1(t-4)} + a_{1t} \\
&= 0,59573Y_{1(t-1)} + 0,12653Y_{1(t-2)} + 0,08164Y_{1(t-3)} + 0,19610Y_{1(t-4)} + a_{1t}
\end{aligned}$$

Persamaan model VARIMA untuk data temperatur di stasiun Semarang menunjukkan bahwa temperatur di stasiun tersebut dipengaruhi oleh temperatur di stasiun yang sama pada 1 hari, 2 hari, 3 hari, dan 4 hari sebelumnya. Sedangkan persamaan model VARIMA untuk data temperatur di stasiun Ahmad Yani adalah sebagai berikut.

$$\begin{aligned}
Y_{2t} &= Y_{2(t-1)} - 0,419Y_{2(t-1)} + 0,419Y_{2(t-2)} - 0,26876Y_{2(t-2)} + 0,26876Y_{2(t-3)} + \\
&\quad 0,18332Y_{1(t-3)} - 0,18332Y_{1(t-4)} - 0,35457Y_{2(t-3)} + 0,35457Y_{2(t-4)} + a_{2t} \\
&= 0,581Y_{2(t-1)} + 0,15024Y_{2(t-2)} - 0,08581Y_{2(t-3)} + 0,18332Y_{1(t-3)} - \\
&\quad 0,18332Y_{1(t-4)} + 0,35457Y_{2(t-4)} + a_{2t}
\end{aligned}$$

Persamaan model VARIMA untuk data temperatur di stasiun Semarang menunjukkan bahwa temperatur di stasiun tersebut dipengaruhi oleh temperatur di stasiun yang sama pada 1 hari, 2 hari, 3 hari, dan 4 hari sebelumnya. Selain itu, juga dipengaruhi oleh temperatur di stasiun Semarang pada 3 hari dan 4 hari sebelumnya.

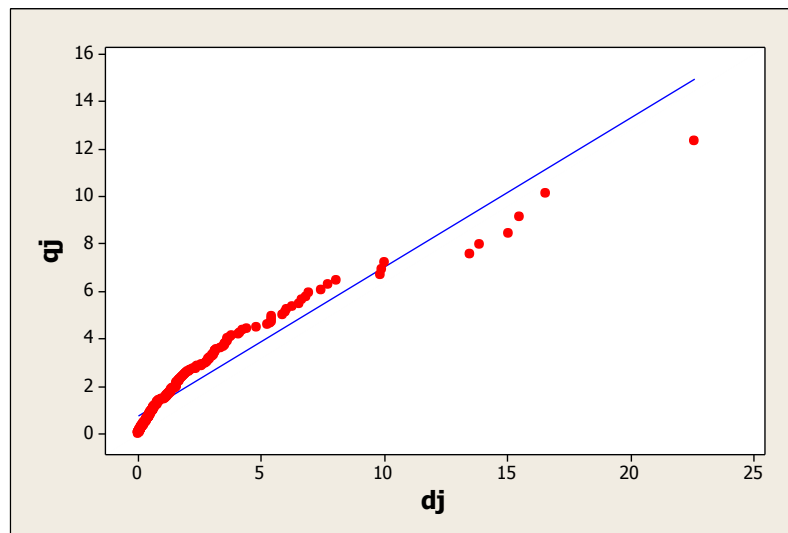
Dari pemodelan tersebut diperoleh nilai MAPE sebesar 0,0201 untuk stasiun Semarang dan 0,0148 untuk stasiun Ahmad Yani. Sedangkan untuk kriteria RMSE masing-masing stasiun diperoleh nilai sebesar 0,6847 dan 0,5115.

Tabel 4.6 Nilai AIC residual model VARIMA (3,1,0)

Lag	MA (0)	MA (1)
AR (0)	-2,057	-2,025
AR (1)	-2,028	-2,044
AR (2)	-2,008	-2,021
AR (3)	-1,990	-1,997
AR (4)	-2,032	-2,040
AR (5)	-2,017	-2,048

Langkah selanjutnya adalah pengujian asumsi apakah model sudah memenuhi asumsi *white noise*. Pengujian dilakukan dengan cara memodelkan ulang residual dari model. Selanjutnya dilakukan pengecekan dimana letak AIC

terkecil. Jika letak AIC terkecil terdapat pada lag AR(0) dan MA(0), maka residual dapat dikatakan sudah memenuhi asumsi *white noise*. Nilai AIC terkecil terletak pada AR(0) dan MA(0) sebesar -2,057 (Tabel 4.6). Hal ini menunjukkan bahwa residual model VARIMA (3,1,0) sudah memenuhi asumsi *white noise*. Selanjutnya, pemeriksaan asumsi normalitas multivariat vektor residual model VARIMA (3,1,0) dengan membuat q-q plot menghasilkan nilai proporsi sebesar 0,61. Artinya bahwa secara subyektif residual sudah memenuhi asumsi normalitas multivariat. Karena nilai proporsi tersebut sudah melebihi setengah dari jumlah pengamatan. Hasil *q-q plot* akan dijelaskan pada Gambar 4.8 sebagai berikut.



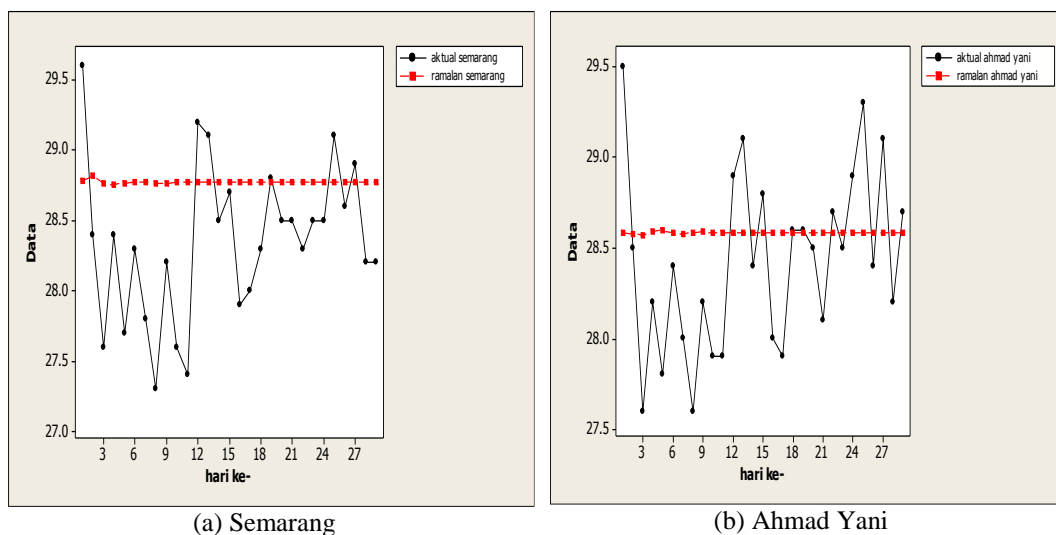
Gambar 4.8 Hasil *q-q plot*

Terdapat beberapa titik pengamatan residual yang menyimpang dari garis lurus (Gambar 4.8). Oleh karena itu, dapat disimpulkan bahwa residual secara visual tidak memenuhi asumsi multivariat normal. Namun, pada penelitian ini akan tetap digunakan sebagai variabel masukan di FFNN.

Hasil ramalan di stasiun Semarang dan Ahmad Yani dengan metode VARIMA (3,1,0) disajikan pada Tabel 4.7. Kedua plot menunjukkan bahwa hasil ramalan dari model VARIMA (3,1,0) menjauhi nilai dari data aktual baik di stasiun Semarang maupun Ahmad Yani (Gambar 4.8). Hasil ramalan cenderung konstan yang berarti bahwa nilai yang diperoleh hampir sama dari hari ke hari sehingga menghasilkan *error* yang cukup tinggi. Oleh karena itu, dilanjutkan pada pemodelan nonlinear dengan metode *Feed Forward Neural Network*.

Tabel 4.7 Hasil ramalan model VARIMA (3,1,0)

Hari ke-	Semarang	Ahmad Yani	Hari ke-	Semarang	Ahmad Yani
1	28,78	28,59	16	28,77	28,58
2	28,82	28,58	17	28,77	28,59
3	28,76	28,57	18	28,77	28,59
4	28,76	28,59	19	28,77	28,59
5	28,77	28,60	20	28,77	28,59
6	28,77	28,58	21	28,77	28,59
7	28,77	28,58	22	28,77	28,59
8	28,77	28,58	23	28,77	28,59
9	28,77	28,59	24	28,77	28,59
10	28,77	28,59	25	28,77	28,59
11	28,77	28,58	26	28,77	28,59
12	28,77	28,58	27	28,77	28,59
13	28,77	28,59	28	28,77	28,59
14	28,77	28,59	29	28,77	28,59
15	28,77	28,59			

**Gambar 4.8** Hasil *timeseries* plot ramalan VARIMA (3,1,0) antara aktual dan ramalan

4.3 Uji Nonlinearitas

Untuk mengetahui ada tidaknya hubungan nonlinier dengan lag-lag yang signifikan maka dilakukan uji nonlinieritas dengan uji Terasvirta. Pemodelan dengan *Neural Network* merupakan salah satu metode analisis yang melibatkan hubungan nonlinier. Oleh karena itu, perlu dilakukan pengujian untuk mendeteksi hubungan nonlinier. Variabel dependen (Y) adalah data temperatur di stasiun Semarang (Y_{1t}) dan Ahmad Yani (Y_{2t}). Sedangkan variabel independen (X) adalah lag-lag yang signifikan pada masing-masing stasiun berdasarkan model VAR. Berikut ini adalah hasil uji Terasvirta disajikan pada Tabel 4.8.

Tabel 4.8 Uji Terasvirta

Y	X	χ^2	p_value	Kesimpulan
Y_{1t}	$Y_{1(t-1)}$	4,3637	0,1128	Linier
	$Y_{1(t-2)}$	7,92	0,01906	Nonlinier
	$Y_{1(t-3)}$	7,0968	0,02877	Nonlinier
Y_{2t}	$Y_{2(t-1)}$	1,4638	0,481	Linier
	$Y_{2(t-2)}$	5,7259	0,0571	Linier
	$Y_{2(t-3)}$	6,2243	0,0445	Nonlinier
	$Y_{1(t-3)}$	6,6387	0,03618	Nonlinier

*) $\alpha=0,05$

Berdasarkan pada Tabel 4.8 maka dapat diketahui bahwa nilai χ^2 merupakan nilai statistik uji Terasvirta yang dibandingkan dengan nilai tabel $\chi^2_{(0,05;2)}$ sebesar 5,991. Apabila nilai χ^2 lebih dari 5,991 atau p_value kurang dari 0,05 maka dapat disimpulkan bahwa terdapat hubungan nonlinier antara variabel Y dan X. Terdapat 4 variabel yang mempunyai hubungan nonlinier. Karena pada masing-masing variabel Y terdapat minimal satu hubungan nonlinier dengan variabel X maka salah satu metode yang dapat digunakan adalah metode *Feed Forward Neural Network*.

4.4 Pemodelan FFNN

Pemodelan *Feed Forward Neural Network* (FFNN) data temperatur di stasiun Semarang dan Ahmad Yani menggunakan enam variabel masukan. Berdasarkan hasil pemodelan VAR keenam variabel tersebut adalah $Y_{1(t-1)}$, $Y_{1(t-2)}$, $Y_{1(t-3)}$, $Y_{2(t-1)}$, $Y_{2(t-2)}$, dan $Y_{2(t-3)}$. Variabel keluaran terdapat dua variabel yaitu Y_{1t} dan Y_{2t} . Arsitektur FFNN yang digunakan adalah satu lapisan tersembunyi dengan jumlah node lapisan tersembunyi (J), dicoba mulai dari $J = 1$ sampai dengan $J = 4$. Fungsi aktivasi antara lapisan masukan dan lapisan tersembunyi $f_1(.)$ yang digunakan adalah fungsi *logistic sigmoid*. Estimasi parameter model dilakukan dengan menggunakan algoritma *backpropagation*. Variabel masukan yang digunakan pertama kali adalah $Y_{1(t-1)}$ dan $Y_{2(t-1)}$ dengan node pada lapisan tersembunyi

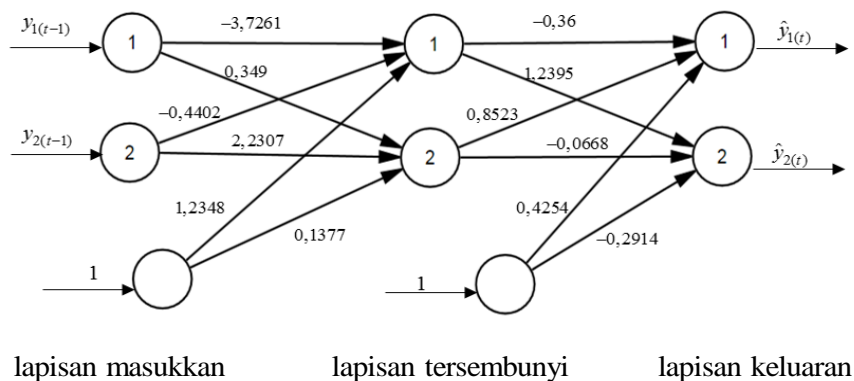
sebanyak satu. Kemudian dengan variabel masukan yang sama dilakukan kembali analisis FFNN dengan node pada lapisan tersembunyi sebanyak 2, 3, dan 4.

Selanjutnya, variabel masukan yang digunakan adalah $Y_{1(t-1)}$, $Y_{2(t-1)}$, $Y_{1(t-2)}$, dan $Y_{2(t-2)}$ dengan node tersembunyi yang dicoba mulai dari 1 sampai dengan 4. Kemudian percobaan yang terakhir menggunakan variabel masukan yaitu $Y_{1(t-1)}$, $Y_{2(t-1)}$, $Y_{1(t-2)}$, $Y_{2(t-2)}$, $Y_{1(t-3)}$, dan $Y_{2(t-3)}$ dengan node tersembunyi yang dicoba mulai dari 1 sampai dengan 4. Untuk memilih arsitektur terbaik, pada data *in-sample* digunakan kriteria AIC dan *out-of-sample* digunakan kriteria RMSE. Nilai AIC untuk model FFNN masing-masing dapat dilihat pada Tabel 4.9 sebagai berikut.

Tabel 4.9 Nilai AIC model FFNN

Model	AIC
FFNN (2,1,2)	-434,2230
FFNN (2,2,2)	-444,6135
FFNN (2,3,2)	-409,4478
FFNN (2,4,2)	-384,1215
FFNN (4,1,2)	-422,0291
FFNN (4,2,2)	-418,0111
FFNN (4,3,2)	-374,8803
FFNN (4,4,2)	-327,9399
FFNN (6,1,2)	-403,7556
FFNN (6,2,2)	-377,5539
FFNN (6,3,2)	-304,6707
FFNN (6,4,2)	-266,166

Model FFNN terbaik berdasarkan kriteria AIC adalah 2 node di lapisan masukan, 2 node di lapisan tersembunyi, dan 2 node di lapisan keluaran atau dapat juga ditulis dengan FFNN (2,2,2) (Tabel 4.9). Bentuk arsitektur FFNN dan estimasi parameter dari model tersebut akan disajikan pada Gambar 4.10 sebagai berikut.



Gambar 4.10 Arsitektur Model FFNN (2,2,2)

Berdasarkan arsitektur FFNN (2,2,2) pada Gambar 4.10 maka persamaan model untuk masing-masing stasiun adalah sebagai berikut.

a. Stasiun Semarang (\hat{Y}_{1t})

$$\hat{Y}_{1t} = 0,4254 - 0,36 f(g_1) + 0,8523 f(g_2)$$

b. Stasiun Ahmad Yani (\hat{Y}_{2t})

$$\hat{Y}_{2t} = -0,2914 + 1,2395 f(g_1) - 0,0668 f(g_2)$$

dimana $f(g_1)$ dan $f(g_2)$ adalah fungsi logistik sigmoid pada lapisan tersembunyi yang didefinisikan sebagai berikut.

$$f(g_1) = \frac{1}{1 + e^{-(g_1)}}$$

$$f(g_2) = \frac{1}{1 + e^{-(g_2)}}$$

dengan g_1 dan g_2 sebagai berikut.

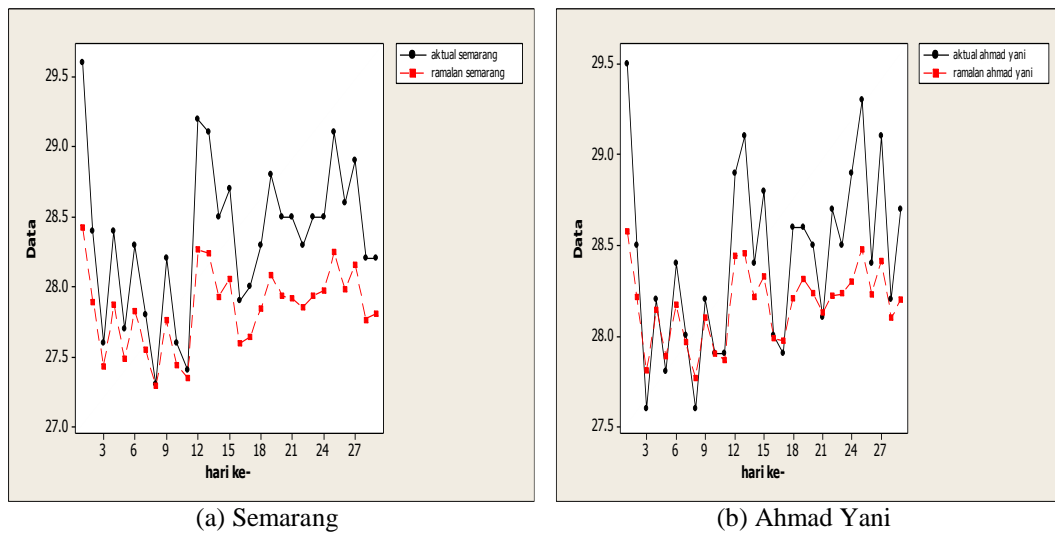
$$g_1 = 1,2348 - 3,7261Y_{1(t-1)} - 0,4402Y_{2(t-1)}$$

$$g_2 = 0,1377 + 0,349Y_{1(t-1)} + 2,2307Y_{2(t-1)}$$

Berdasarkan model di atas maka dapat diperoleh hasil ramalan di stasiun Semarang dan Ahmad Yani yang disajikan pada Tabel 4.10 sebagai berikut.

Tabel 4.10 Hasil ramalan model FFNN (2,2,2)

Hari ke-	Semarang	Ahmad Yani	Hari ke-	Semarang	Ahmad Yani
1	28,4	28,6	16	27,6	28,0
2	27,9	28,2	17	27,6	28,0
3	27,4	27,8	18	27,8	28,2
4	27,9	28,1	19	28,1	28,3
5	27,5	27,9	20	27,9	28,2
6	27,8	28,2	21	27,9	28,1
7	27,5	28,0	22	27,9	28,2
8	27,3	27,8	23	27,9	28,2
9	27,8	28,1	24	28,0	28,3
10	27,4	27,9	25	28,3	28,5
11	27,3	27,9	26	28,0	28,2
12	28,3	28,4	27	28,2	28,4
13	28,2	28,5	28	27,8	28,1
14	27,9	28,2	29	27,8	28,2
15	28,1	28,3			



Gambar 4.11 Hasil *timeseries* plot ramalan FFNN (2,2,2) antara aktual dan ramalan

Kedua plot menunjukkan bahwa hasil ramalan dari model FFNN (2,2,2) mendekati nilai dari data aktual (Gambar 4.11). Hasil ramalan model FFNN lebih baik dibandingkan model VAR. Kriteria kebaikan model menggunakan MAPE diperoleh hasil sebesar 0,0175 untuk stasiun Semarang dan 0,0103 untuk stasiun Ahmad Yani. Sedangkan kriteria RMSE menunjukkan nilai untuk masing-masing stasiun sebesar 0,5643 dan 0,3879.

4.5 Pemeriksaan Distribusi Data

Sebelum dilakukan analisis menggunakan pendekatan Bayesian untuk model FFNN maka terlebih dahulu dilakukan pemeriksaan distribusi. Data temperatur di stasiun Semarang dan Ahmad Yani diduga berdistribusi normal sehingga hipotesis yang digunakan adalah sebagai berikut.

H_0 : Data temperatur berdistribusi normal

H_1 : Data temperatur tidak berdistribusi normal

Hasil pengujian distribusi normal pada data temperatur di stasiun Semarang dan Ahmad Yani ditampilkan pada Tabel 4.11 sebagai berikut.

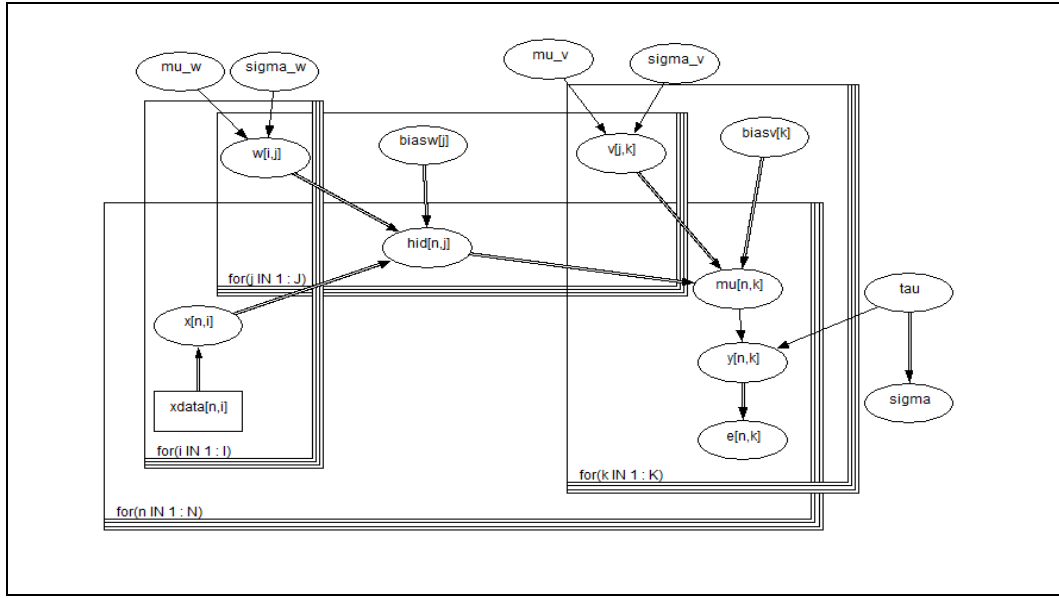
Tabel 4.11 Pemeriksaan distribusi data				
Stasiun	Statistik	Nilai kritis	p_value	Distribusi
Semarang	0,06756	0,08712	0,20767	Normal
Ahmad Yani	0,0449	0,08712	0,69398	Normal

Hasil pemeriksaan distribusi data temperatur di stasiun Semarang dan Ahmad Yani dengan menggunakan *Kolmogorov-Smirnov* diperoleh nilai sebesar 0,06756 dan 0,0449 (Tabel 4.11). Dimana nilai statistik tersebut kurang dari nilai kritis dengan α sebesar 0,05 atau nilai p_value lebih besar dari α . Sehingga dapat disimpulkan bahwa data temperatur di stasiun Semarang dan Ahmad Yani telah mengikuti distribusi normal.

4.6 Pemodelan Bayesian *Neural Network*

Setelah diperoleh distribusi data untuk masing-masing stasiun maka langkah selanjutnya adalah melakukan pemodelan Bayesian *Neural Network* (BNN). Model BNN menggunakan fungsi aktivasi pada lapisan tersembunyi adalah logistik sigmoid, sedangkan fungsi aktivasi pada lapisan keluaran menggunakan fungsi linier. Hasil analisis pemeriksaan distribusi data disimpulkan bahwa likelihood yang digunakan dalam penelitian ini adalah normal sehingga $y_i \sim N(\mu_i, \tau_i)$. Parameter-parameter yang akan diestimasi adalah parameter w dan v . Pada analisis bayesian digunakan *hyperparameter* pada prior sehingga parameter pada v berdistribusi multivariat normal dimana $w \sim N_p(\mu_w, \tau_w)$ dengan μ_w berdistribusi multivariat normal dan τ_w berdistribusi wishart. Begitu pula dengan parameter v berdistribusi multivariat normal dimana $v \sim N_p(\mu_v, \tau_v)$ dengan μ_v berdistribusi multivariat normal dan τ_v berdistribusi wishart.

Proses *generate sample* pada tiap parameter yang diestimasi harus random pada analisis metode bayesian dengan WinBugs (Ntzoufras, 2009). Salah satu cara untuk mengatasi adanya autokorelasi pada hasil *generate sample* tiap parameter adalah dengan menggunakan *thin* sesuai dengan hasil identifikasi pada *autocorrelation function* (ACF) WinBugs. Apabila proses masih tetap tidak random maka terdapat kemungkinan terjadi kekompleksan model yang mengakibatkan WinBugs kesulitan dalam menjaga konsistensi random sampel. *Doodle* WinBugs untuk memodelkan FFNN dengan pendekatan bayesian akan dijelaskan pada Gambar 4.12 sebagai berikut.



Gambar 4.12 Doodle WinBugs model BNN

Pemodelan FFNN dengan menggunakan metode optimasi Levenberg Marquadt diperoleh hasil bahwa model terbaik adalah FFNN (2,2,2). Oleh karena itu, pemodelan BNN digunakan model lengkap. Sehingga model BNN diperoleh model terbaik adalah model BNN dengan menggunakan bias atau konstanta pada parameter w dan v .

Gambar 4.12 adalah *doodle* WinBugs untuk analisis BNN. Sehingga diperoleh *code* WinBugs dan dilakukan sedikit modifikasi untuk prior dan inisialisasi parameter. Selanjutnya mencari prior yang tepat dalam penelitian ini menggunakan *hyperparameter*. Proses menentukan prior yang tepat dilakukan dengan cara pertama menggunakan prior untuk matrik varians kovarians adalah matrik identitas dan matriks rata-rata bernilai nol sebagai berikut.

$$\mathbf{w}_{ij} \sim N\left(\boldsymbol{\mu}_{\mathbf{w}_{ij}}, \boldsymbol{\Sigma}_{\mathbf{w}_{ij}}\right) \quad ; i = 1, 2 \quad j = 1, 2$$

$$\text{dimana } \mathbf{w}_{ij} = \begin{bmatrix} w_{11} \\ w_{12} \\ w_{21} \\ w_{22} \end{bmatrix}$$

$$\boldsymbol{\mu}_{\mathbf{w}_{ij}} \sim N(\mathbf{a}_{\mathbf{w}_{ij}}, \mathbf{B}_{\mathbf{w}_{ij}}^{-1})$$

$$\text{dimana } \mathbf{a}_{\mathbf{w}_{ij}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{B}_{\mathbf{w}_{ij}}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Sigma_{\mathbf{w}_{ij}}^{-1} \sim \text{Wishart}(\mathbf{A}_{\mathbf{w}_{ij}}, b_{\mathbf{w}_{ij}}) \quad \text{dimana } \mathbf{A}_{\mathbf{w}_{ij}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{v}_{jk} \sim N(\boldsymbol{\mu}_{\mathbf{v}_{jk}}, \Sigma_{\mathbf{v}_{jk}}) \quad ; j=1,2 \quad k=1,2$$

$$\text{dimana } \mathbf{v}_{jk} = \begin{bmatrix} v_{11} \\ v_{12} \\ v_{21} \\ v_{22} \end{bmatrix}$$

$$\boldsymbol{\mu}_{\mathbf{v}_{jk}} \sim N(\mathbf{a}_{\mathbf{v}_{jk}}, \mathbf{B}_{\mathbf{v}_{jk}}^{-1})$$

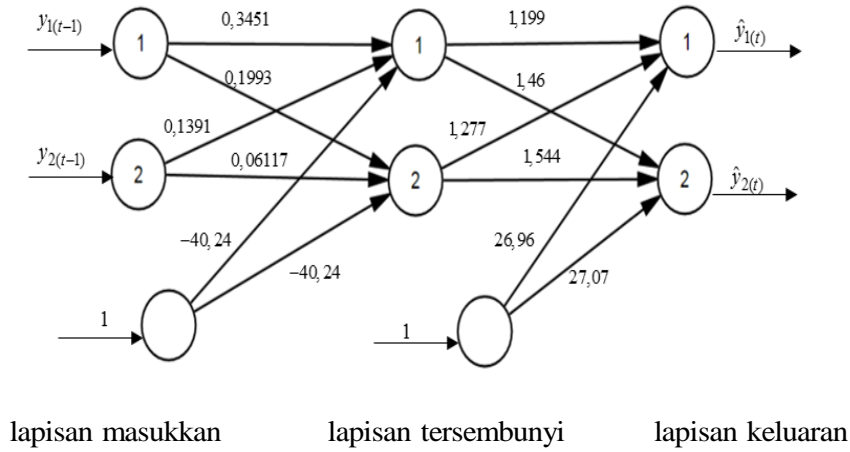
$$\text{dimana } \mathbf{a}_{\mathbf{v}_{jk}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{B}_{\mathbf{v}_{jk}}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Sigma_{\mathbf{v}_{jk}}^{-1} \sim \text{Wishart}(\mathbf{A}_{\mathbf{v}_{jk}}, b_{\mathbf{v}_{jk}}) \quad \text{dimana } \mathbf{A}_{\mathbf{v}_{jk}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Berdasarkan distribusi prior di atas maka akan dilakukan estimasi parameter BNN (2,2,2) dimana lapisan masukan yang digunakan adalah 2 variabel yaitu $Y_{1(t-1)}$ dan $Y_{2(t-1)}$ dengan dua node di lapisan tersembunyi dan 2 node di lapisan keluaran. Nilai taksiran parameter diperoleh dari distribusi *posterior* dimana proses pengambilan sampel parameter dilakukan dengan menggunakan metode MCMC dan *Gibbs Sampling*.

Pada penelitian ini prosedur MCMC dilakukan 5000 iterasi dengan *thin* pada 100 iterasi dan *burn-in* sebanyak 200 untuk mendapatkan sampel parameter dari distribusi *posterior* sehingga total sampel yang digunakan adalah 4801 sampel. Setelah proses pengambilan sampel dilakukan, maka langkah selanjutnya adalah

diagnosa untuk mengetahui kesesuaian hasil estimasi dengan sifat-sifat *Markov Chain* yang *strongly ergodic* dan konvergen. Bentuk arsitektur BNN dan estimasi parameter dengan pendekatan bayesian akan disajikan pada Gambar 4.12 sebagai berikut.



Gambar 4.13 Arsitektur Model Bayesian *Neural Network*

Berdasarkan arsitektur BNN (2,2,2) pada Gambar 4.13 maka persamaan model untuk masing-masing stasiun adalah sebagai berikut.

a. Stasiun Semarang (\hat{Y}_{1t})

$$\hat{Y}_{1t} = 26,96 + 1,199 f(g_1) + 1,277 f(g_2)$$

b. Stasiun Ahmad Yani (\hat{Y}_{2t})

$$\hat{Y}_{2t} = 27,07 + 1,46 f(g_1) + 1,544 f(g_2)$$

dimana $f(g_1)$ dan $f(g_2)$ adalah fungsi logistik sigmoid pada lapisan tersembunyi yang didefinisikan sebagai berikut.

$$f(g_1) = \frac{1}{1 + e^{-(g_1)}}$$

$$f(g_2) = \frac{1}{1 + e^{-(g_2)}}$$

dengan g_1 dan g_2 sebagai berikut.

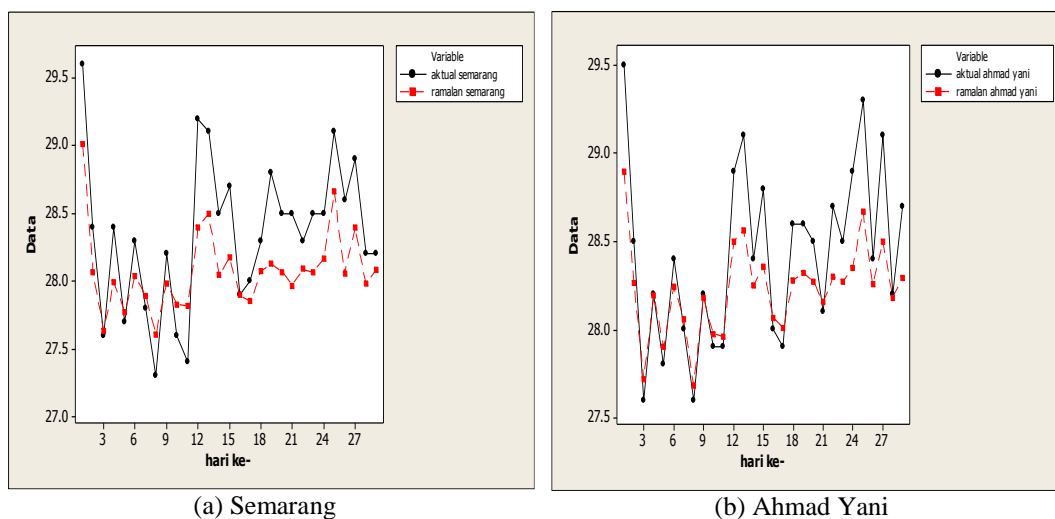
$$g_1 = -40,24 + 0,3451Y_{1(t-1)} + 0,1391Y_{2(t-1)}$$

$$g_2 = -40,24 + 0,1993Y_{1(t-1)} + 0,06117Y_{2(t-1)}$$

Berdasarkan model di atas maka dapat diperoleh hasil ramalan di stasiun Semarang dan Ahmad Yani yang disajikan pada Tabel 4.12 sebagai berikut.

Tabel 4.12 Hasil ramalan model BNN

Hari ke-	Semarang	Ahmad Yani	Hari ke-	Semarang	Ahmad Yani
1	29.2	29.8	16	27.9	28.2
2	28.1	28.5	17	27.9	28.2
3	27.5	27.8	18	28.2	28.5
4	28.1	28.4	19	28.2	28.6
5	27.7	28.0	20	28.2	28.5
6	28.1	28.5	21	28.0	28.4
7	27.9	28.2	22	28.2	28.6
8	27.5	27.7	23	28.2	28.5
9	28.0	28.4	24	28.3	28.7
10	27.8	28.1	25	28.8	29.3
11	27.8	28.1	26	28.1	28.5
12	28.5	29.0	27	28.5	29.0
13	28.6	29.1	28	28.0	28.4
14	28.1	28.5	29	28.2	28.5
15	28.3	28.7			



Gambar 4.13 Hasil *timeseries* plot ramalan BNN antara aktual dan ramalan

Hasil dari *timeseries* plot model BNN (2,2,2) menunjukkan bahwa hasil ramalan lebih baik dibandingkan metode VAR (Gambar 4.13). Berdasarkan pemodelan Bayesian *Neural Network* maka diperoleh nilai AIC sebesar -407,0369. Sedangkan untuk kriteria *out-of-sample* menggunakan MAPE diperoleh hasil sebesar 0,0096 untuk stasiun Semarang dan 0,0050 untuk stasiun Ahmad Yani.

Hasil RMSE menunjukkan nilai untuk masing-masing stasiun sebesar 0,3195 dan 0,1651.

Pada analisis BNN dapat dilakukan evaluasi signifikansi parameter-parameter. Pemeriksaan signifikansi parameter dilakukan dengan menggunakan batas bawah dan atas adalah nilai persentil 2,5% dan 97,5%. Hasil nilai parameter-parameter model BNN (2,2,2) adalah sebagai berikut.

Tabel 4.13 Hasil nilai parameter-parameter model BNN

node	mean	sd	2,5%	97,5%
bias_v [1]	26,75	1,125	23,85	29,09
bias_v [2]	26,88	1,065	24,31	29,45
bias_w [1]	-40,24	21,99	-98,72	-14,58
bias_w [2]	-125,8	862,3	-1932	1733
sigma	0,6308	0,04231	0,5539	0,7185
v [1,1]	1,907	0,951	0,03794	3,24
v [1,2]	2,054	0,6051	0,8466	3,106
v [2,1]	0,9997	5,368	-4,434	4,857
v [2,2]	0,7398	5,381	-3,946	4,75
w [1,1]	0,5242	0,5681	-0,7062	1,296
w [1,2]	0,4678	1,986	-3,677	3,291
w [2,1]	0,9106	0,9405	-0,07535	3,397
w [2,2]	-0,04709	1,738	-3,378	3,3

Nilai mean pada tiap parameter di Tabel 4.13 menunjukkan estimasi parameter dengan bayesian untuk tiap bobot pada model FFNN. Sedangkan nilai persentil 2,5% dan 97,5% digunakan untuk mengevaluasi signifikansi bobot-bobot pada model FFNN. Apabila selang interval memuat angka nol maka dapat disimpulkan bahwa parameter tidak signifikan. Jadi, dapat disimpulkan bahwa hampir semua parameter tidak signifikan karena memuat angka nol (Tabel 4.13). Parameter yang signifikan hanya v_{11} dan v_{12} sehingga perlu dilakukan *improve*.

Langkah selanjutnya dilakukan pemeriksaan hasil dari *autocorrelation function* (ACF) pada tiap parameter. Langkah ini dilakukan untuk mengevaluasi *generate sample* dalam proses MCMC pada tiap parameter telah terambil secara acak atau belum. Apabila proses masih belum acak, maka perlu ditambahkan nilai *thin*. Hasil ACF dan *time series* plot dijelaskan pada Lampiran 9 dan 10. Karena hasil ACF dan *time series* plot belum memenuhi sifat markov maka dilakukan *improve* dengan memasukkan nilai *mean* dan standar deviasi yang diperoleh dari Tabel 4.13. Nilai *mean* pada tiap parameter di Tabel 4.14 menunjukkan estimasi parameter dengan bayesian untuk tiap bobot pada model FFNN setelah *improve*.

Jadi, dapat disimpulkan bahwa semua parameter tidak signifikan karena memuat angka nol (Tabel 4.14). Hasil ACF dan *time series* plot dijelaskan pada Lampiran 11 dan 10. Beberapa parameter sudah memenuhi sifat markov dan menunjukkan hasil yang lebih baik (Lampiran 10 dan 11).

Tabel 4.14 Hasil nilai parameter-parameter model BNN setelah *improve*

node	mean	sd	2,5%	97,5%
bias_v [1]	26.96	0.2486	26.45	27.28
bias_v [2]	27.07	0.3059	26.04	27.39
bias_w [1]	-40.24	0.2118	-40.65	-39.82
bias_w [2]	-40.24	0.215	-40.66	-39.83
sigma	0.6319	0.04166	0.5563	0.7191
v [1,1]	1.199	2.683	-3.867	3.66
v [1,2]	1.46	1.493	-1.336	2.61
v [2,1]	1.277	5.628	-4.19	4.164
v [2,2]	1.544	3.146	-1.29	2.804
w [1,1]	0.3451	1.862	-3.064	2.031
w [1,2]	0.1993	0.7578	-1.436	1.03
w [2,1]	0.1391	1.628	-2.678	2.379
w [2,2]	0.06117	0.8038	-2.104	0.8502

4.7 Perbandingan Hasil Berdasarkan Kriteria Model Terbaik

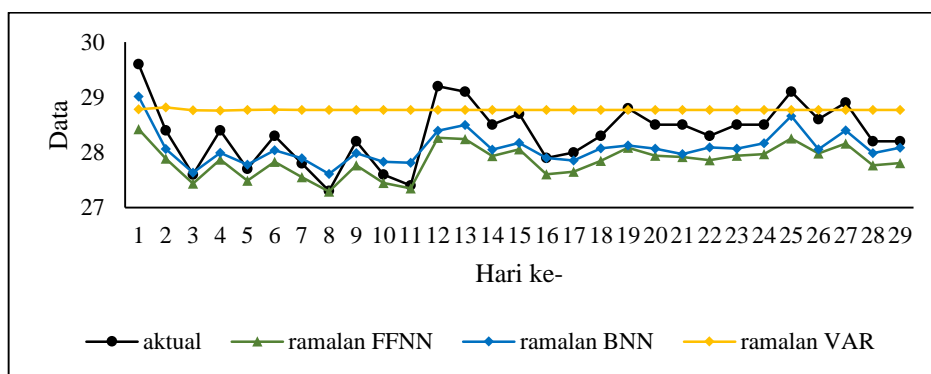
Berdasarkan hasil analisis sebelumnya, data temperatur di kota Semarang dianalisis dengan menggunakan *Vector Autoregressive* (VAR), *Feed Forward Neural Network* (FFNN) dan *Bayesian Neural Network* (BNN). Hasil analisis pemodelan dan peramalan diperoleh nilai kriteria kebaikan model yang berbeda-beda. Tabel 4.15 merupakan hasil perbandingan antara ketiga model berdasarkan kriteria *in-sample* dan *out-of-sample*.

Tabel 4.15 Hasil perbandingan model

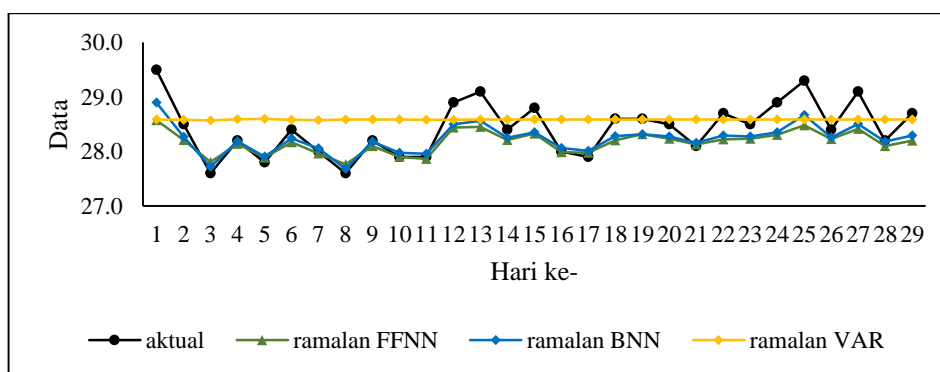
Model	Kriteria <i>in-sample</i>		Kriteria <i>out-of-sample</i>		
	AIC	MAPE		RMSE	
		Semarang	Ahmad Yani	Semarang	Ahmad Yani
VARIMA (3,1,0)	-1,95679	0,0201	0,0148	0,6847	0,5115
FFNN (2,2,2)	-444,6135	0,0175	0,0103	0,5643	0,3879
BNN (2,2,2)	-407,0369	0,0096	0,0050	0,3195	0,1651

Hasil analisis pada pemodelan linier yaitu VAR dan nonlinier yaitu FFNN dan BNN menunjukkan bahwa pemodelan nonlinier lebih baik daripada pemodelan linier. Karena nilai AIC, MAPE dan RMSE pada metode FFNN dan BNN lebih kecil daripada metode VAR. Jadi, dapat disimpulkan bahwa model BNN mempunyai kinerja lebih baik daripada VAR dan FFNN (Tabel 4.15). Hasil

perbandingan *timeseries plot* secara visual dari ketiga model menunjukkan bahwa hasil ramalan dengan BNN lebih banyak yang mendekati data aktual daripada model FFNN (Gambar 4.15). Oleh karena itu, dapat disimpulkan bahwa hasil ramalan model BNN lebih baik dibandingkan dengan VAR dan FFNN.



(a) Semarang



(b) Ahmad Yani

Gambar 4.15 Hasil perbandingan *timeseries plot* data aktual dan ramalan

Model terbaik berdasarkan kriteria *out-of-sample* adalah model BNN sehingga hasil ramalan 7 periode ke depan adalah sebagai berikut.

Tabel 4.15 Hasil nilai ramalan temperatur harian dengan metode BNN

Hari ke-	Semarang	Ahmad Yani
1	27,61	27,83
2	27,22	27,32
3	27,26	27,45
4	27,50	27,61
5	27,39	27,51
6	27,68	27,88
7	27,38	27,39

LAMPIRAN

Lampiran 1. Data Temperatur Semarang Tahun 2013

Data <i>in-sample</i>				Data <i>out-of-sample</i>			
Bulan	Hari ke-	Semarang	Ahmad Yani	Bulan	Hari ke-	Semarang	Ahmad Yani
Januari	1	27,9	27,6	September	1	29,6	29,5
	2	26,6	27,1		2	28,4	28,5
	3	26,9	26,9		3	27,6	27,6
	⋮	⋮	⋮		⋮	⋮	⋮
	31	27,4	27,7		30	28,4	28,3
Februari	1	27,7	28,2				
	2	27,8	28,1				
	3	28,2	28,4				
	⋮	⋮	⋮				
	28	28,1	28,7				
Maret	1	28,2	28,8				
	2	27,6	28,2				
	3	28,0	28,9				
	⋮	⋮	⋮				
	31	28,5	28,6				
⋮	⋮	⋮	⋮				
Agustus	1	27,8	27,3				
	2	27,4	27,3				
	3	27,5	27,2				
	⋮	⋮	⋮				
	31	28,7	28,6				

Lampiran 2. Macro SAS untuk VAR

```
data temperatur;
input y1 y2;
datalines;
27.9    27.6
26.6    27.1
26.9    26.9
.
29.1    28.8
28.6    28.5
28.7    28.6
.
;
proc varmax data=temperatur lagmax=24 printall;
model y1-y2 / p=3 dify(1) dfptest noint print=(corry pcorr);
restrict AR(1,1,2)=0 AR(3,1,2)=0 AR(2,2,1)=0 AR(2,1,2)=0 AR(1,2,1)=0;
output lead=29 out=hasil;
run;

proc print data=hasil;
run;
```

Lampiran 3. Hasil Estimasi Parameter Model VARIMA (3,1,0)

Lampiran 3.1 Sebelum *Restrict*

Model Parameter Estimates						
Equation	Parameter	Estimate	Standard Error	t Value	Pr > t	Variable
y1	AR1_1_1	-0.46487	0.11152	-4.17	0.0001	y1(t-1)
	AR1_1_2	0.01692	0.10888	0.16	0.8767	y2(t-1)
	AR2_1_1	-0.18127	0.11707	-1.55	0.1229	y1(t-2)
	AR2_1_2	-0.1248	0.1131	-1.1	0.271	y2(t-2)
	AR3_1_1	-0.22111	0.11104	-1.99	0.0476	y1(t-3)
	AR3_1_2	0.01912	0.10872	0.18	0.8605	y2(t-3)
y2	AR1_2_1	-0.1087	0.11362	-0.96	0.3397	y1(t-1)
	AR1_2_2	-0.35849	0.11093	-3.23	0.0014	y2(t-1)
	AR2_2_1	0.07507	0.11928	0.63	0.5297	y1(t-2)
	AR2_2_2	-0.36934	0.11523	-3.21	0.0015	y2(t-2)
	AR3_2_1	0.1587	0.11313	1.4	0.162	y1(t-3)
	AR3_2_2	-0.33641	0.11076	-3.04	0.0027	y2(t-3)

Lampiran 3.2 Setelah *Restrict*

Model Parameter Estimates						
Equation	Parameter	Estimate	Standard Error	t Value	Pr > t	Variable
y1	AR1_1_1	-0.40427	0.04816	-8.39	0.0001	y1(t-1)
	AR1_1_2	0	0			y2(t-1)
	AR2_1_1	-0.27774	0.05092	-5.46	0.0001	y1(t-2)
	AR2_1_2	0	0			y2(t-2)
	AR3_1_1	-0.1961	0.06143	-3.19	0.0016	y1(t-3)
	AR3_1_2	0	0			y2(t-3)
y2	AR1_2_1	0	0			y1(t-1)
	AR1_2_2	-0.419	0.04789	-8.75	0.0001	y2(t-1)
	AR2_2_1	0	0			y1(t-2)
	AR2_2_2	-0.26876	0.05015	-5.36	0.0001	y2(t-2)
	AR3_2_1	0.18332	0.07715	2.38	0.0183	y1(t-3)
	AR3_2_2	-0.35457	0.05977	-5.93	0.0001	y2(t-3)

Testing of the Restricted Parameters				
Parameter	Estimate	Standard Error	t Value	Pr > t
AR1_1_2	-6.52398	13.43648	-0.49	0.6277
AR1_2_1	-15.6921	12.88771	-1.22	0.2246
AR2_1_2	-10.6857	13.06753	-0.82	0.4144
AR2_2_1	3.15065	12.21258	0.26	0.7966
AR3_1_2	5.64659	9.71424	0.58	0.5616

Lampiran 4. Uji Multivariat Normal

```
macro
qq x.1-x.p
mconstant i n p t chis
mcolumn d x.1-x.p dd pi q ss tt y.1-y.p
mmatrix s sinv ma mb mc md
let n=count(x.1)
cova x.1-x.p s
invert s sinv

do i=1:p
  let y.i=x.i-mean(x.i)
enddo

do i=1:n
  copy y.1-y.p ma;
  use i.
  transpose ma mb
  multiply ma sinv mc
  multiply mc mb md
  copy md tt
  let t=tt(1)
  let d(i)=t
enddo

set pi
  1:n
end

let pi=(pi-0.5)/n
sort d dd
invcdf pi q;
chis p.
plot q*dd
invcdf 0.5 chis;
chis p.
let ss=dd<chis
let t=sum(ss)/n
print t

endmacro
```

Lampiran 5. Uji Terasvirta Menggunakan Program R

```
library(tseries)
kasus<-read.csv("D:/temperatursemarang.csv",sep=";",header=TRUE)
semarang<-kasus$Semarang
ahmadyani<-kasus$Ahmadyani
y1t<-semarang[4:243]
y1t1<-semarang[3:242]
y1t2<-semarang[2:241]
y1t3<-semarang[1:240]
y2t<-ahmadyani[4:243]
y2t1<-ahmadyani[3:242]
y2t2<-ahmadyani[2:241]
y2t3<-ahmadyani[1:240]
y2_1t3<-semarang[1:240]
terasvirta.test(x1,y1t)
terasvirta.test(x2,y2t)
terasvirta.test(y1t1,y1t)
terasvirta.test(y1t2,y1t)
terasvirta.test(y1t3,y1t)
terasvirta.test(y2t1,y2t)
terasvirta.test(y2t2,y2t)
terasvirta.test(y2t3,y2t)
terasvirta.test(y2_1t3,y2t)

> terasvirta.test(y1t1,y1t)
      Teraesvirta Neural Network Test
data: y1t1 and y1t
X-squared = 4.3637, df = 2, p-value = 0.1128

> terasvirta.test(y1t2,y1t)
      Teraesvirta Neural Network Test
data: y1t2 and y1t
X-squared = 7.92, df = 2, p-value = 0.01906
```

```
> terasvirta.test(y1t3,y1t)
      Teraesvirta Neural Network Test
data: y1t3 and y1t
X-squared = 7.0968, df = 2, p-value = 0.02877

> terasvirta.test(y2t1,y2t)
      Teraesvirta Neural Network Test
data: y2t1 and y2t
X-squared = 1.4638, df = 2, p-value = 0.481

> terasvirta.test(y2t2,y2t)
      Teraesvirta Neural Network Test
data: y2t2 and y2t
X-squared = 5.7259, df = 2, p-value = 0.0571

> terasvirta.test(y2t3,y2t)
      Teraesvirta Neural Network Test
data: y2t3 and y2t
X-squared = 6.2243, df = 2, p-value = 0.0445

> terasvirta.test(y2_1t3,y2t)
      Teraesvirta Neural Network Test
data: y2_1t3 and y2t
X-squared = 6.6387, df = 2, p-value = 0.03618
```


Lampiran 6. Mfile Matlab FFNN (2,2,2)

```
%panggil data
clc
datatraining=xlsread('datatraining.xls');
datatesting=xlsread('datatesting.xls');
%inputkan data
tdata=datatraining';
lag1=tdata(:,1:242);
insample=[lag1];
target=tdata(:,2:243);
tout=datatesting';
outlag1=tout(:,1:29);
outsample=[outlag1];
targetoutput=tout(:,2:30);
i=0;
AIC_opt=-1;
i_opt_train=0;

for i=0:99;
    i=i+1
    %penentuan banyaknya node pada hidden dan output serta insialisasi bobot
    net=newff(insample,target,[2 2],{'logsig' 'purelin'},'trainlm');
    bobotawalinput=net.IW{1,1}
    bobotawalbiasinput=net.b{1,1}
    bobotawaloutput=net.LW{2,1}
    bobotawalbiasoutput=net.b{2,1}

    %penentuan epoch dan error
    net.trainParam.epochs=1000;
    net.trainParam.goal=1e-3;
    net.trainParam.lr=0.01;

    %penentuan bobot akhir
    net=train(net,insample,target);
    bobotakhirinput=net.IW{1,1}
    bobotakhirbiasinput=net.b{1,1}
    bobotakhiroutput=net.LW{2,1}
    bobotakhirbiasoutput=net.b{2,1}

    %ramalan insample dan perhitungan residual
    y_insample=sim(net,insample);
    y_in=[y_insample(1,:)';y_insample(2,:)'];
    ramalanin(:,i)=y_in(:,1)
    e=(target-y_insample);
    resi=e';
    residual=[e(1,:)';e(2,:)'];
    e_kuadrat=residual.^2;
    varcov_in=cov(resi);
    det_in=det(varcov_in);
    [o n]=size(insample);
```

```

%jumlah parameter
p_input=size(bobotakhirinput,1)*size(bobotakhirinput,2);
p_biasinput=size(bobotakhirbiasinput,1)*size(bobotakhirbiasinput,2);
p_output=size(bobotakhiroutput,1)*size(bobotakhiroutput,2);
p_biasoutput=size(bobotakhirbiasoutput,1)*size(bobotakhirbiasoutput,2);
p=p_input+p_biasinput+p_output+p_biasoutput;
[m k]=size(targetoutput);

%kriteria kebaikan model in
mse_insample=mean(e_kuadrat);
rmse_insample=sqrt(mse_insample);
A_insample=n*log(det_in)+2*p*(m^2)
AIC(i)=A_insample(1)
y_outsample=sim(net,outsample)
yout=[y_outsample(1,:)' ;y_outsample(2,:)]
ramalanout(:,i)=yout(:,1)
e_out=targetoutput-y_outsample;
residual_out=[e_out(1,:)' ;e_out(2,:)]';
e_outkuadrat=residual_out.^2;
    if(A_insample <= AIC_opt)
        AIC_opt=A_insample;
        i_opt_train=i;
        IW_opt=net.IW{1,1}
        b1_opt=net.b{1,1}
        LW_opt=net.LW{2,1}
        b2_opt=net.b{2,1}

%ramalan outsample
y_outsample=sim(net,outsample)
yout=[y_outsample(1,:)' ;y_outsample(2,:)]
ramalanout(:,i)=yout(:,1);
e_out=targetoutput-y_outsample;
residual_out=[e_out(1,:)' ;e_out(2,:)]';
e_outkuadrat=residual_out.^2;
end;

disp('---hasil terbaik---')
disp 'AIC_opt'
disp (AIC_opt);
disp 'i_opt_train'
disp (i_opt_train);
disp 'IW_opt'
disp (IW_opt);
disp 'b1_opt'
disp (b1_opt);
disp 'LW_opt'
disp (LW_opt);
disp 'b2_opt'
disp (b2_opt);
disp 'ramalan_opt'
disp (ramalanout);
end

```

Lampiran 7. Pemeriksaan Distribusi Data

Lampiran 7.1 Pemeriksaan Distribusi Normal data temperatur di stasiun Semarang

Normal [#44]					
Kolmogorov-Smirnov					
Sample Size	243				
Statistic	0.06756				
P-Value	0.20767				
Rank	12				
α	0.2	0.1	0.05	0.02	0.01
Critical Value	0.06883	0.07846	0.08712	0.09738	0.1045
Reject?	No	No	No	No	No

Lampiran 7.2 Pemeriksaan Distribusi Normal data temperatur di stasiun Ahmad yani

Normal [#44]					
Kolmogorov-Smirnov					
Sample Size	243				
Statistic	0.0449				
P-Value	0.69398				
Rank	8				
α	0.2	0.1	0.05	0.02	0.01
Critical Value	0.06883	0.07846	0.08712	0.09738	0.1045
Reject?	No	No	No	No	No

Lampiran 8. Syntax WinBugs Bayesian Neural Network

```
model;
{
for( n in 1 : N ) {
for( k in 1 : K ) {
y[n , k] <- ydata[n , k]
}
}
for( n in 1 : N ) {
for( i in 1 : I ) {
x[n , i] <- xdata[n , i]
}
}

#menentukan prior
mu_w[1:J] ~ dmnorm(meanmu_w[ ],taumu_w[ , ])
tau_w[1:J , 1:J] ~ dwish(sigma_w[ , ], J)
mu_v[1:K] ~ dmnorm(meanmu_v[ ],taumu_v[ , ])
tau_v[1:K , 1:K] ~ dwish(sigma_v[ , ], K)

for( i in 1 : I ) {
w[i , 1:J] ~ dmnorm(mu_w[ ],tau_w[ , ])
}

for( j in 1 : J ) {
v[j , 1:K] ~ dmnorm(mu_v[ ],tau_v[ , ])
bias_w[j]~ dnorm(0.0,1.0E-6)

}

for (k in 1:K){
bias_v[k] ~ dnorm(0.0,1.0E-6)
}
tau ~ dgamma( 0.001,0.001)
sigma <- 1 / tau

#menghitung prediksi
for( n in 1 : N ) {
for( j in 1 : J ) {
hid[n , j] <- 1 / (1 + exp(-bias_w[j]-inprod(w[ , j],x[n , ])))
}
}
for( n in 1 : N ) {
for( k in 1 : K ) {
mu[n , k] <- bias_v[k]+inprod(v[ , k],hid[n , ])
y[n , k] ~ dnorm(mu[n , k],tau)
e[n , k] <- y[n , k] - mu[n , k]
}
}
}
```

```

#data
list(I=2, J=2, K=2, N=242,
meanmu_w = c(0,0),
meanmu_v = c(0,0),
taumu_w = structure(.Data = c(1,0,0,1), .Dim = c(2, 2)),
sigma_w = structure(.Data = c(1,0,0,1), .Dim = c(2, 2)),
taumu_v = structure(.Data = c(1,0,0,1), .Dim=c(2,2)),
sigma_v = structure(.Data = c(1,0,0,1), .Dim=c(2,2)),
ydata=structure(.Data = c(
26.6 ,      27.1   ,
26.9 ,      26.9   ,
27.3 ,      27.5   ,
:          :
29.1 ,      28.8   ,
28.6 ,      28.5   ,
28.7 ,      28.6   ), .Dim=c(242,2)),
xdata = structure( .Data = c(
27.9 ,      27.6   ,
26.6 ,      27.1   ,
26.9 ,      26.9   ,
:          :
29.0 ,      28.6   ,
29.1 ,      28.8   ,
28.6 ,      28.5   ), .Dim=c(242,2)))

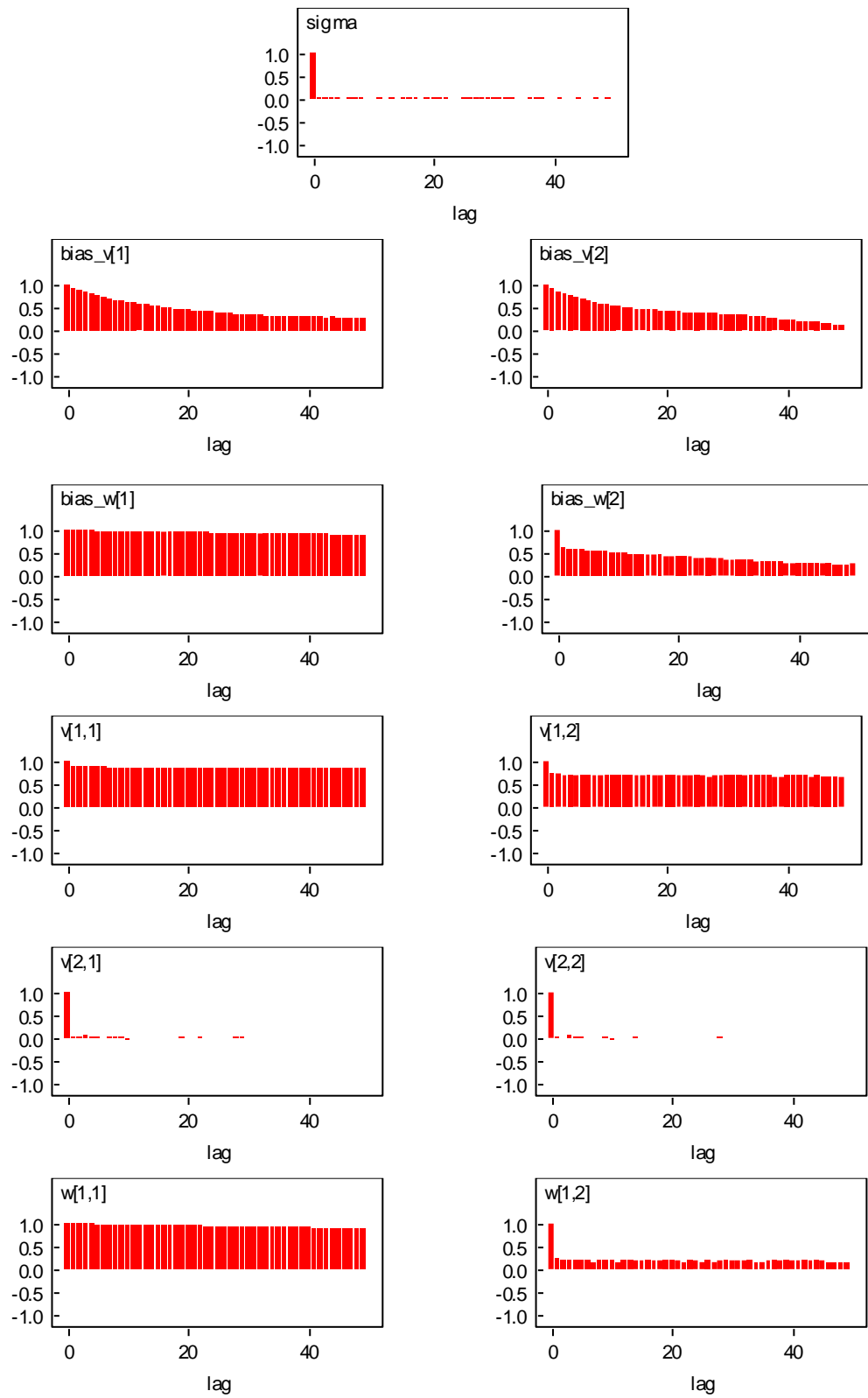
# nilai inisial
list(tau=1, mu_w=c(0,0), mu_v=c(0,0),

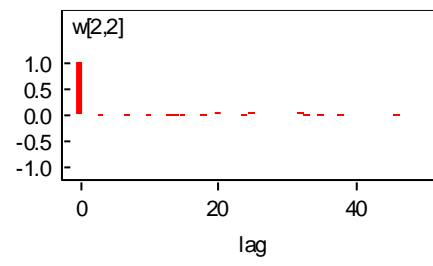
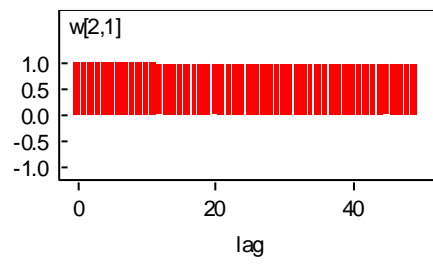
tau_w=structure(.Data = c(1,0,0,1), .Dim=c(2,2)),
tau_v=structure(.Data = c(1,0,0,1), .Dim=c(2,2)),

v=structure(.Data = c(
0,0,
0,0), .Dim=c(2,2)),
w = structure(.Data = c(
0,0,
0,0), .Dim=c(2,2)))

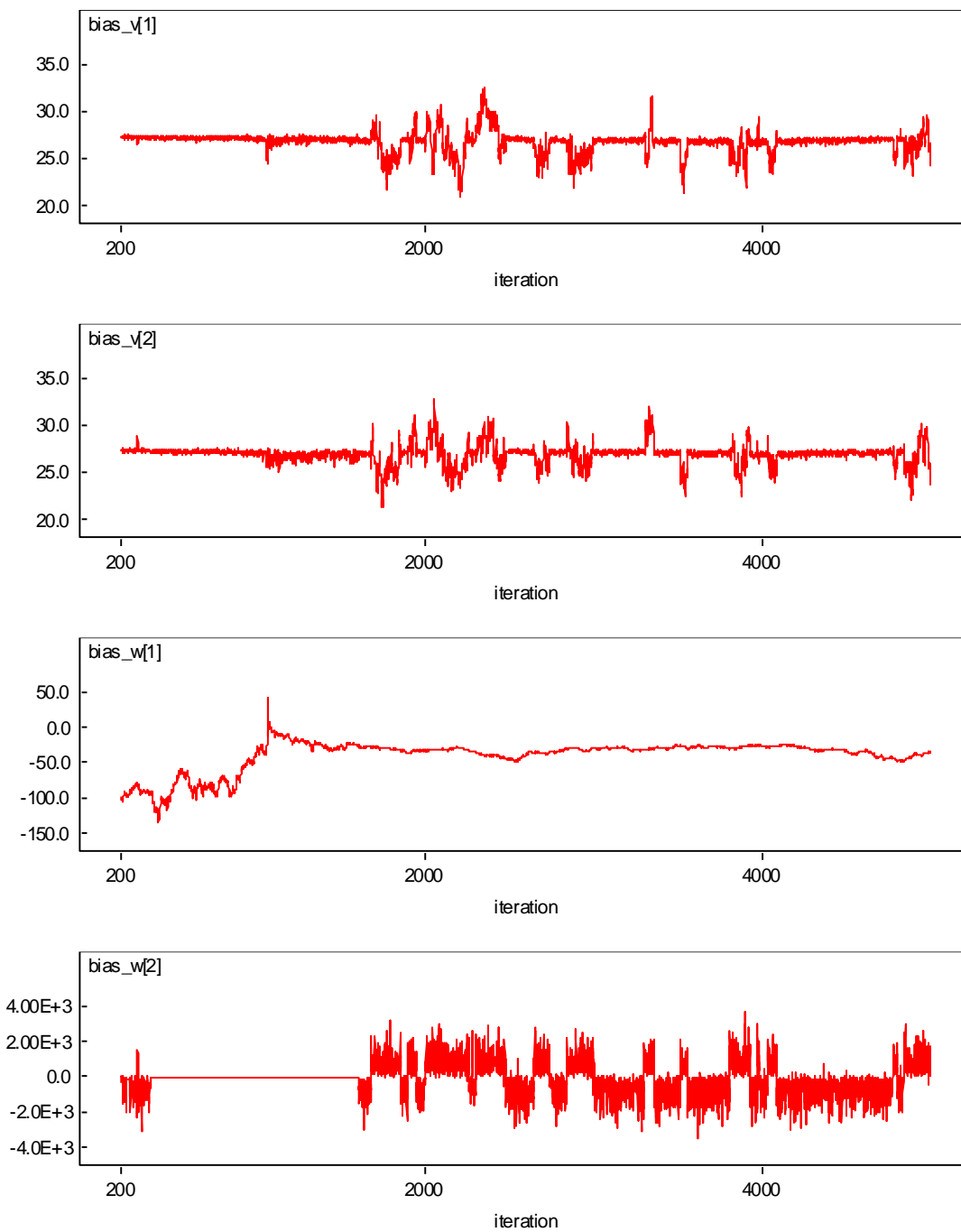
```

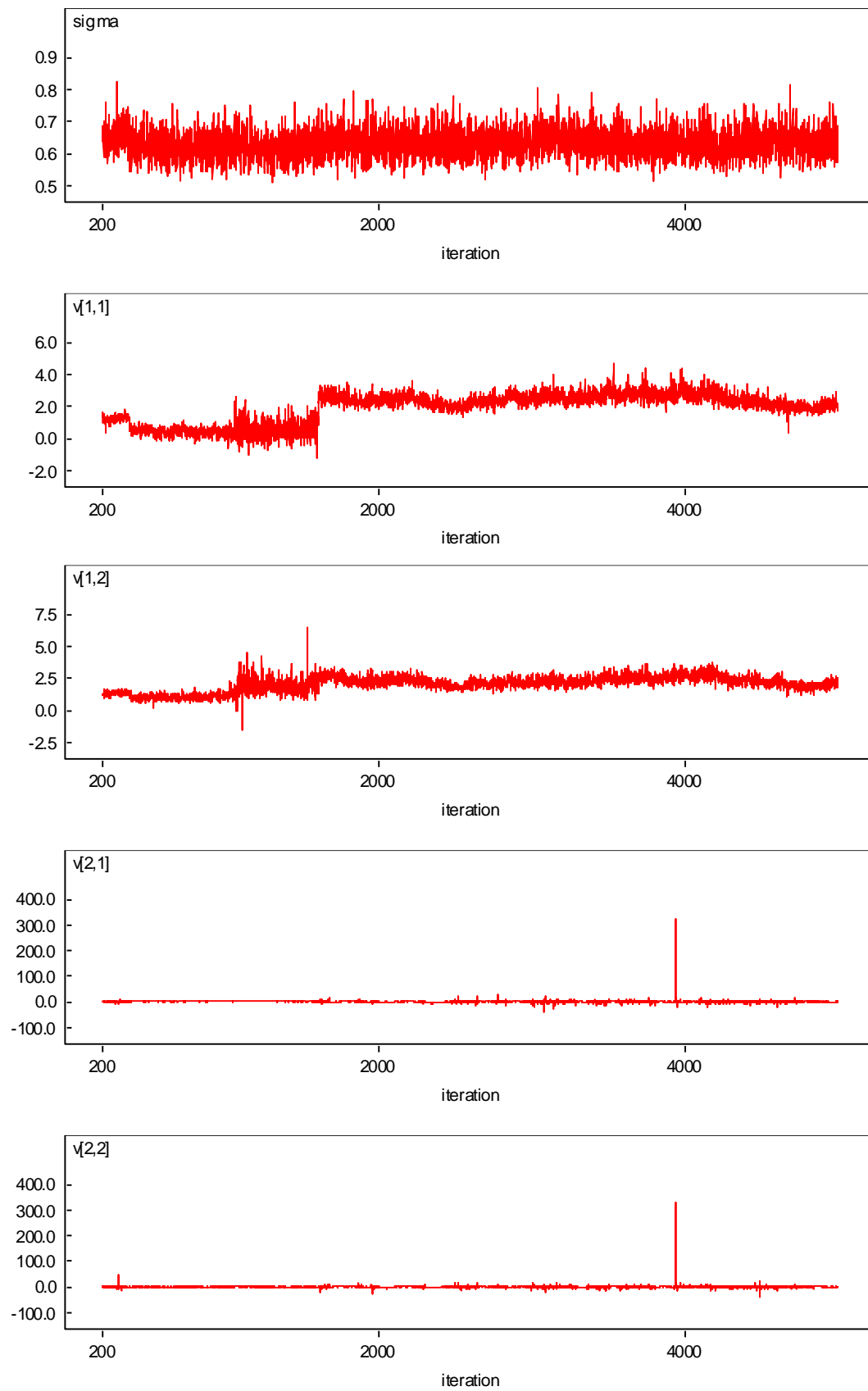
Lampiran 9. ACF BNN

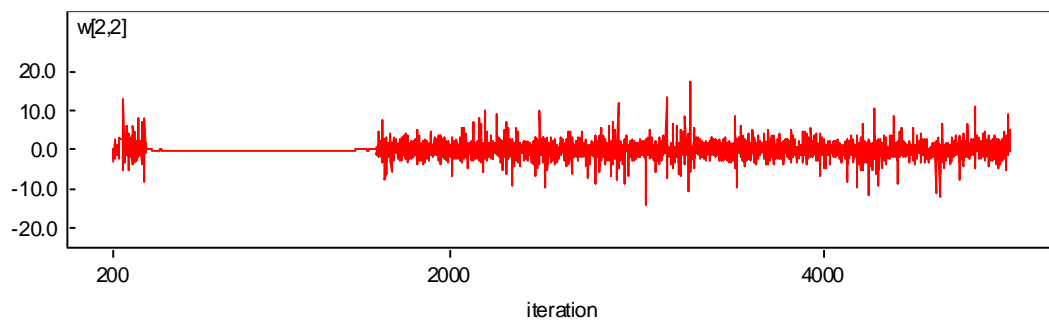
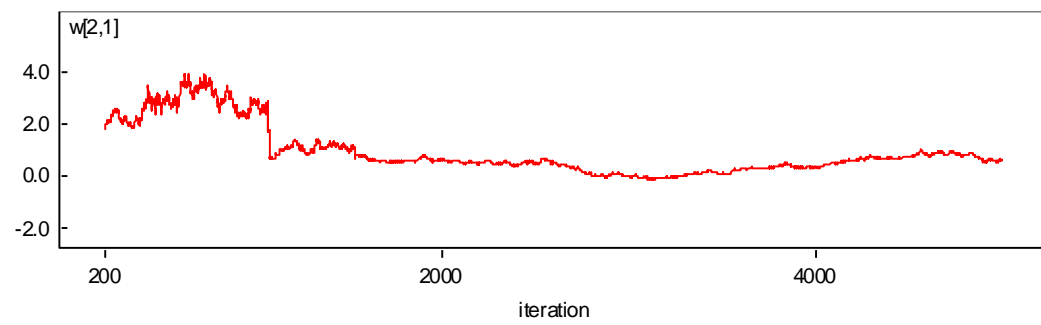
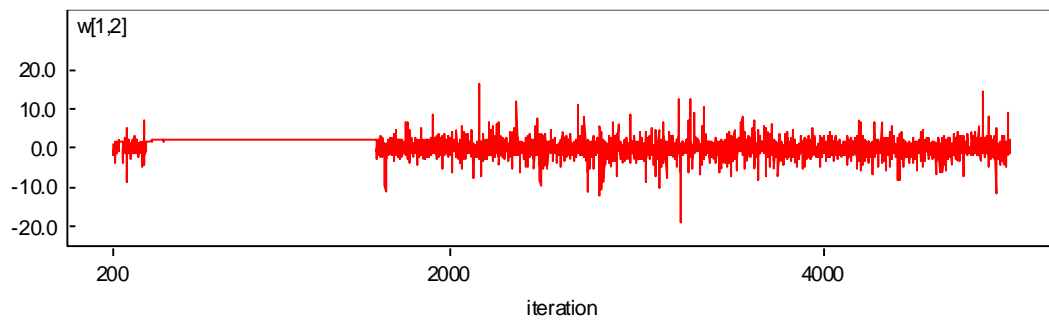




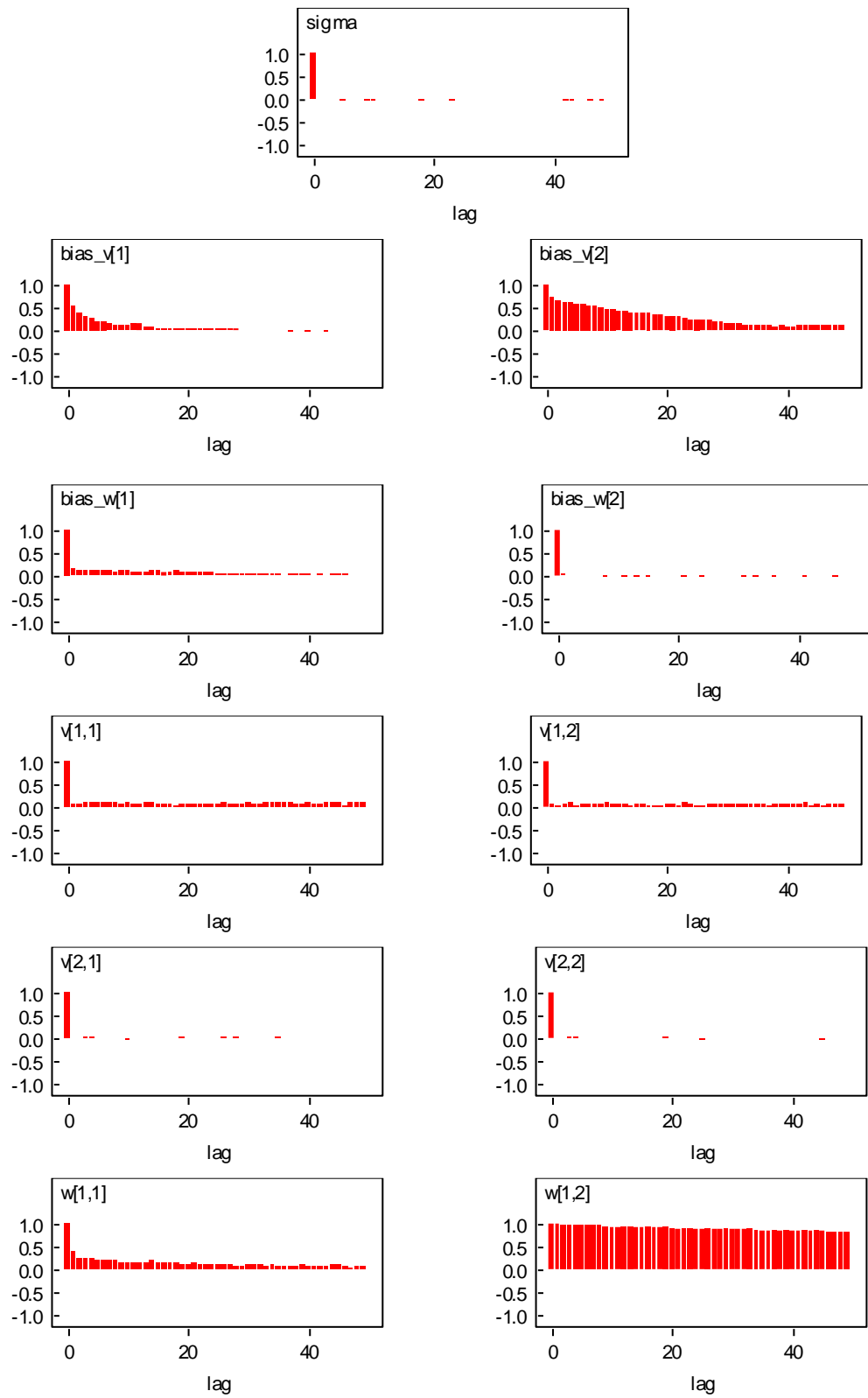
Lampiran 10. Time series plot parameter-parameter BNN

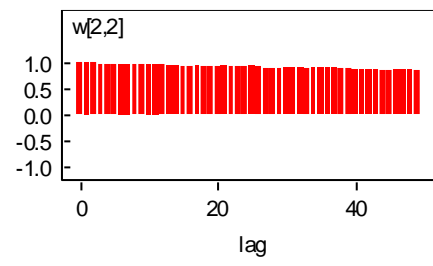
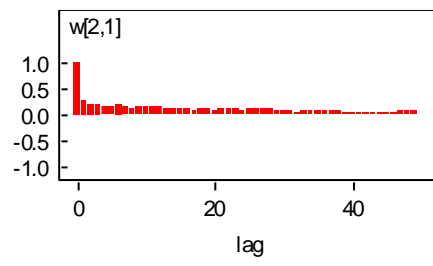




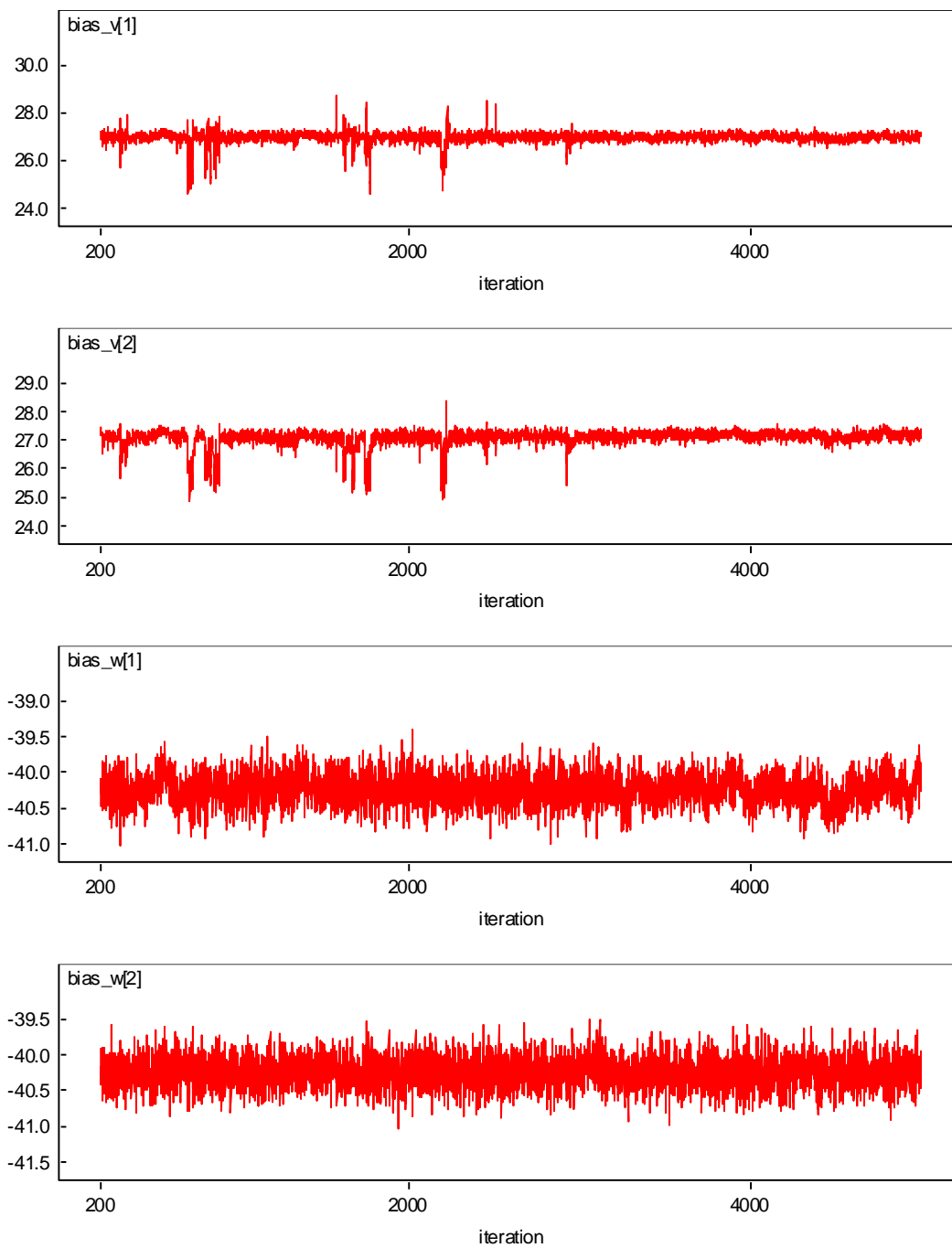


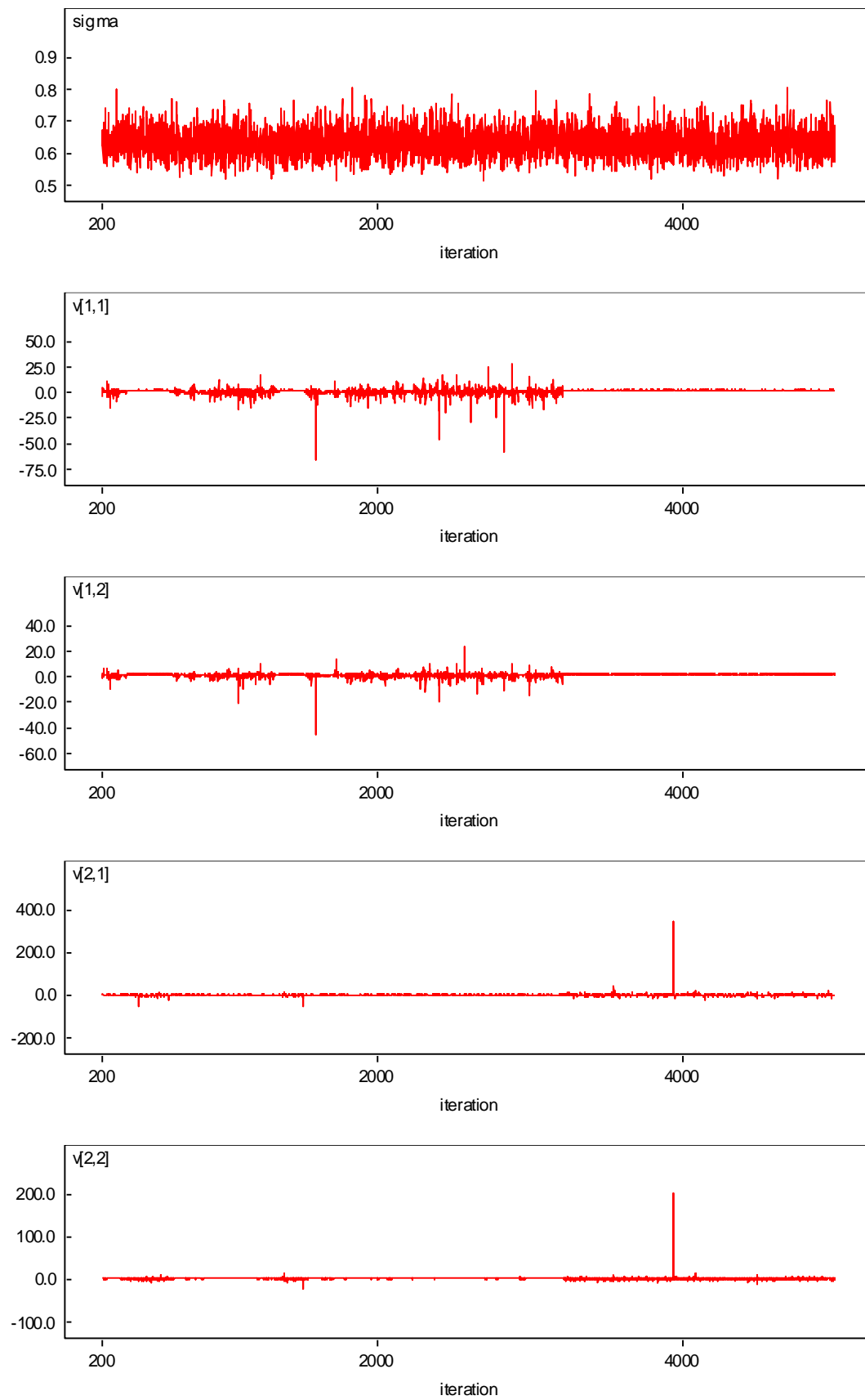
Lampiran 11. ACF BNN setelah *improve*

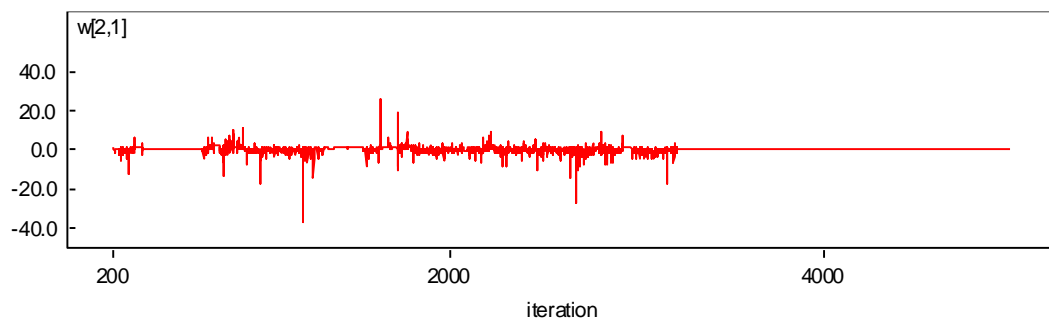
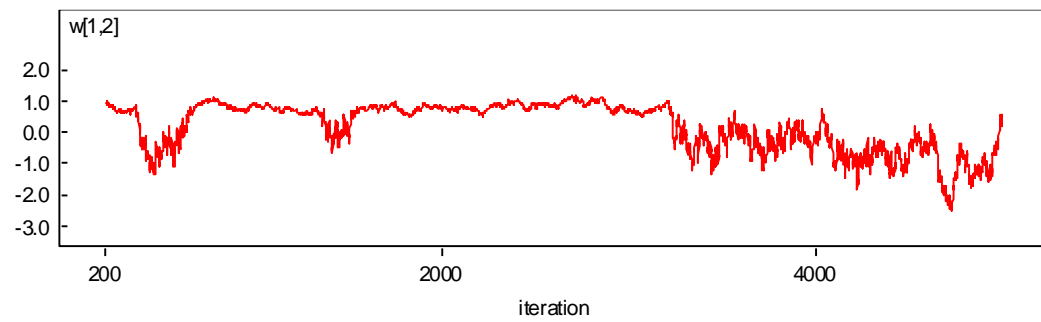
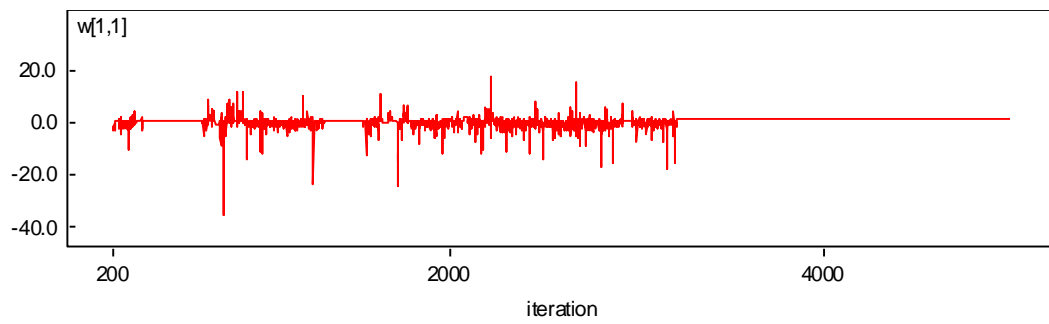




Lampiran 12. Time series plot parameter-parameter BNN setelah *improve*







(Halaman ini sengaja dikosongkan)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis dan pembahasan pada bab sebelumnya maka diperoleh kesimpulan sebagai berikut.

1. Model FFNN terbaik yang digunakan untuk memodelkan temperatur di Semarang adalah FFNN (2,2,2) yang terdiri dari 2 node pada lapisan masukan, 2 node pada lapisan tersembunyi dan 2 node pada lapisan keluaran.
2. Hasil perbandingan kriteria model terbaik kasus temperatur di Semarang menunjukkan bahwa pemodelan BNN mempunyai kinerja yang lebih baik daripada VAR dan FFNN untuk kriteria *out-of sample*. Sedangkan model FFNN mempunyai kinerja lebih baik daripada VAR dan BNN berdasarkan kriteria *in-sample*.

5.2 Saran

Pada analisis Bayesian *Neural Network* masih terdapat autokorelasi yang berarti bahwa *generate sample* pada tiap parameter masih belum acak. Sehingga untuk penelitian selanjutnya sebaiknya mencari prior yang lebih baik lagi. Model BNN pada penelitian ini masih terbatas satu distribusi pada lokasi yang berbeda. Oleh karena itu, perlu dikembangkan model mixture dengan komponen mixture yang berdistribusi berbeda di setiap lokasi.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- Bishop, C. M. (1995). *Neural Network for Pattern Recognition*. UK: Clarendon Press Oxford.
- Box, G., & Tiao, G. (1992). *Bayesian inference in statistical analysis*. New York: Wiley.
- Carlin, B. (1992). A simple Monte Carlo approach to Bayesian graduation. *Transactions of the Society of Actuaries*, 44, 55-76.
- Casella, G., & George, E. (1992). Explaining the Gibbs sampler. *The American Statistician*, 46(3), 167-174.
- Charlin, B. P., & Chib, S. (1995). Bayesian Model via Markov Chain Monte Carlo Methods. *Journal Royal Statistical Society*, 57, No. 3, 473-484.
- Doan, C. D., & Liong, S. Y. (2004). Generalization For Multilayer Neural Network Bayesian Regularization or Early Stopping. *Asia Pasific Association of Hydrology and Water Resources : The 2nd APHW Conference*.
- Fithriasari, K. (2014). *Pengembangan Disagregasi Data Curah Hujan Dengan Pendekatan Bayesian Mixture Neural Network Multi-Output*. Surabaya: Disertasi, Jurusan Statistika, ITS.
- Fithriasari, K., Iriawan, N., Ulama, B., & Sutikno. (2013). Bayesian Neural Networks for Time Series Modeling. *Proceedings Of South East Asian Conference On Mathematics And Its Applications (SEACMA)*. ISBN 978-979-96152-8-2.
- Fithriasari, K., Iriawan, N., Ulama, B., Sutikno, & Kuswanto, H. (2013). Prediction of hourly rainfall using Bayesian neural network adjusting procedure. *Proceeding Of The 3rd Annual, Basic Science International Conference* (hal. 193-201). ISSN 0973-1377(Print), ISSN 0973-7545(Online).
- Gujarati. (2004). *Basic Econometrics Fourth Edition*. New York: The McGraw-Hill.
- IPCC. (2007). *Climate Change 2007: Synthesis Report, Contribution of Working Groups I, II, and III to the Fourth Assessment Report of the Intergovernment Panel on Climate Change*. Geneva, Switzerland: IPCC.
- Iriawan, N., Astutik, S., & Prasetyo, D. D. (2010). Markov Chain Monte Carlo-Based Approaches for Modeling the Spatial Survival with Conditional Autoregressive (CAR) Fraity. *International Journal of Computer Sciences and Network Security*, Vol. 10 No. 12, 211-216.
- Johnson, R. A., & Winchern, D. W. (2007). *Applied Multivariate Statistical Analysis 6th edition*. New Jersey: Pearson Prentice Hall.

- King, R., Morgan, B., Giemenez, O., & Brooks, S. (2010). *Bayesian Analysis for Population Ecology*. USA: Chapman & Hall/CRC.
- Kisi, O., & Uncuoglu, E. (2005). Comparison of Three Back-Propagation Training Algorithms for Two Case Studies. *Indian Journal of Engineering & Materials Sciences*, Vol.12, 434-442.
- Koop, G. (2003). *Bayesian Econometrics*. England: John Wiley & Sons.
- Kvalevag, M. M. (2009). Climate Effects of Land Use Changes And Anthropogenic Impact On Surface Radiation. *ISSN 1501-7710*. Norway: University of Oslo.
- Lampinen, J., & Vehtari, A. (2001). Bayesian approach for neural networks-Review and case studies. *Neural Networks*, 14(3), 7-24.
- Lee, T. H., White, H., & Granger, C. W. (1993). Testing For Neglected Nonlinearity in Time Series Models. *Journal of Econometrics*, 269-290.
- LIPI. (2014, November Senin). Diambil kembali dari <http://kependudukan.lipi.go.id/id/kajian-kependudukan/desa-kota/176-perubahan-kualitas-udara-peningkatan-suhu-di-semarang-dan-bandung>
- Lutkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis*. New York: Springer.
- Machmudin, A., & Ulama, B. S. (2012). Peramalan Temperatur Udara di Kota Surabaya Dengan Menggunakan ARIMA Dan Artificial Neural Network. *Jurnal Sain dan Seni ITS* Vol.1 No.1, D118-D123.
- Menchero, A., Diez, R. M., & Insua, D. R. (2005). Bayesian Analysis of Non-Linear Autoregression Models Based on Neural Networks. *Neural Computation*, 17, 453-485.
- Muller, P., & Insua, D. R. (1998). Issue in Bayesian Analysis of Neural Network Models, Vol 10. *Neural Computation*, 571-592.
- Neal, R. M. (1995). *Bayesian Learning for Neural Network*. University of Toronto.
- Ntzoufras, I. (2009). *Bayesian Modelling Using WinBUGS*. New Jersey, USA: John Wiley & Sons.
- Permai, D. S. (2013). Bayesian Neural Network Untuk Peramalan Temperatur Di Surabaya. *Prosiding Seminar Nasional Pasca Sarjana XIII*, ISBN no. 978-979-96700-6-9. Surabaya: ITS.
- Purnomo, H., & Rizal. (2000). Pengaruh Kelembaban, Temperatur Udara dan Beban Kerja terhadap Kondisi Faal Tubuh Manusia. *Logika*, Vol.4 No.5, ISSN: 1410-2315, 35-47.
- Samadi. (2006). *Geografi*. Yogyakarta: Yudhistira.

- Scollnik, D. M. (2011). Handbook of Markov Chain Monte Carlo Chapter 1. Dalam *An Introduction to Markov Chain Monte Carlo Methods and Their Acturial Applications* (hal. 114-165). New York: Chapman & Hall.
- Siang, J. J. (2005). *Aplikasi Jaringan Syaraf Tiruan dan Pemrograman Menggunakan Matlab*. Yogyakarta: Andi Yogyakarta.
- Sivanandam, A. S., Sumathi, S., & Deepa, S. N. (2006). *Introduction to Neural Network Using Matlab 6.0*. New Delhi: McGraw Hill.
- Suhartono. (2007). *Feed Forward Neural Network untuk Pemodelan Runtun Waktu*. Yogyakarta: Universitas Gajah Mada.
- Terasvirta, T., Lin, C. F., & Granger, C. J. (1993). Power of The Neural Network Linearity Test. *Journal of Time Series Analysis Vol. 14, No. 2*, 159-171.
- Tursilowati. (2007). Urban Heat Island Dan Kontribusinya Pada Perubahan Iklim Dan Hubungannya Dengan Perubahan Lahan. *Prosiding Seminar Nasional Pemanasan Global dan Perubahan Global: Fakta, Mitigasi, dan Adaptasi*.
- Wei, W. S. (2006). *Time Series Analysis: Univariate and Multivariate Method Second Edition*. United States of America: Pearson Education, Inc.
- Wikipedia. (2015, Oktober Minggu). Diambil kembali dari https://id.wikipedia.org/wiki/Kota_Semarang
- Yu, H., & Wilamowski, B. M. (2010). Fast and Efficient and Training of Neural Network. *Proc. 3rd IEEE Human System Interaction Conf. HSI 2010*, (hal. 15-22). Rzeszow, Poland.

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Penulis dengan nama lengkap Zubdatu Zahrati lahir di kota Surabaya pada tanggal 05 Juli 1990, anak ketiga dari tiga bersaudara. Penulis melanjutkan pendidikan pada jenjang S-2 di Statistika ITS pada tahun 2014. Penulis juga pernah menjadi asisten dosen pada mata kuliah analisis multivariat, biostat, dan pengendalian kualitas statistik. Penulis berharap hasil penelitian tugas akhir ini dapat bermanfaat bagi pembaca. Apabila pembaca ingin berdiskusi mengenai laporan Tesis ini, dapat menghubungi melalui email: zubdatu.zahrati05@gmail.com.